Capitaliser les processus d'analyse de traces d'apprentissage : modélisation ontologique & assistance à la réutilisation

Alexis Lebis

Sorbonne Université

École Doctorale Informatique, Télécommunication et Électronique de Paris

Laboratoire LIP6 (UMR 7606) / Équipe MOCAH
Laboratoire LIRIS (UMR 5205) / Équipe TWEAK

Capitaliser les processus d'analyse de traces d'apprentissage

Modélisation ontologique & Assistance à la réutilisation

Par **Alexis Lebis**

Thèse de doctorat en Informatique

Dirigée par **Vanda Luengo**, **Marie Lefevre** & **Nathalie Guin**

Présentée et soutenue publiquement le 22 mai 2019

Devant un jury composé de :

M. **Michel C. DESMARAIS**,
Professeur titulaire. École polytechnique Montréal. Rapporteur.
Mme. **Catherine FARON-ZUCKER**,
Maître de conférences HDR. Polytech'Nice-Sophia. Rapporteur.
M. **Serge GARLATTI**,
Professeur des universités. Lab-STICC, IMT Atlantique. Examinateur.
M. **Christophe MARSALA**,
Professeur des universités. Sorbonne Université. Examinateur.
M. **Julien BROISIN**,
Maître de conférences. Université Paul Sabatier. Examinateur.
Mme. **Vanda LUENGO**,
Professeur des universités. Sorbonne Université. Directrice de thèse.
Mme. **Marie LEFEVRE**,
Maître de conférences. Université Claude Bernard Lyon 1. Co-encadrante.
Mme. **Nathalie GUIN**,
Maître de conférences HDR. Université Claude Bernard Lyon 1. Co-encadrante.

Résumé

Cette thèse en informatique porte sur la problématique de la capitalisation des processus d'analyse de traces d'apprentissage au sein de la communauté des Learning Analytics (LA). Il s'agit de permettre de partager, adapter et réutiliser ces processus d'analyse de traces.

Actuellement, cette capitalisation est limitée par deux facteurs importants : les processus d'analyse sont dépendants des outils d'analyse qui les mettent en œuvre - leur contexte technique - et du contexte pédagogique pour lequel ils sont menés. Cela empêche de les partager, mais aussi de les réexploiter simplement en dehors de leurs contextes initiaux, quand bien même les nouveaux contextes seraient similaires.

L'objectif de cette thèse est de fournir des modélisations et des méthodes permettant la capitalisation des processus d'analyse de traces d'apprentissage, ainsi que d'assister les différents acteurs de l'analyse, notamment durant la phase de réutilisation. Pour cela, nous répondons aux trois verrous scientifiques suivant : comment partager et combiner des processus d'analyse mis en œuvre dans différents outils d'analyse ? ; comment permettre de réexploiter un processus d'analyse existant pour répondre à un autre besoin d'analyse ? ; comment assister les différents acteurs lors de l'élaboration et de l'exploitation de processus d'analyse ?

Notre première contribution, issue d'une synthèse de l'état de l'art, est la formalisation d'un cycle d'élaboration et d'exploitation des processus d'analyse, afin d'en définir les différentes étapes, les différents acteurs et leurs différents rôles. Cette formalisation est accompagnée d'une définition de la capitalisation et de ses propriétés.

Notre deuxième contribution répond au premier verrou lié à la dépendance technique des processus d'analyse actuels, et à leur partage. Nous proposons un méta-modèle qui permet de décrire les processus d'analyse indépendamment des outils d'analyse. Ce méta-modèle formalise la description des opérations utilisées dans les processus d'analyse, des processus eux-mêmes et des traces utilisées, afin de s'affranchir des contraintes techniques occasionnées par ces outils. Ce formalisme commun aux processus d'analyse permet aussi d'envisager leur partage. Il a été mis en œuvre et évalué dans un de nos prototypes.

Notre troisième contribution traite le deuxième verrou sur la réexploitation des processus d'analyse. Nous proposons un framework ontologique pour les processus d'analyse, qui permet d'introduire de manière structurée des éléments sémantiques dans la description des processus d'analyse. Cette approche narrative enrichit ainsi le formalisme précédent et permet de satisfaire les propriétés de compréhension, d'adaptation et de réutilisation nécessaires à la capitalisation. Cette approche ontologique a été mise en œuvre et évaluée dans un autre de nos prototypes.

Enfin, notre dernière contribution répond au dernier verrou identifié et concerne de nouvelles pistes d'assistances aux acteurs, notamment une nouvelle méthode de recherche des processus d'analyse, s'appuyant sur nos propositions précédentes. Nous exploitons le cadre ontologique de l'approche narrative pour définir des règles d'inférence et des heuristiques permettant de raisonner sur les processus d'analyse dans leur ensemble (e.g. étapes, configurations) lors de la recherche. Nous utilisons également le réseau sémantique sous-jacent à cette modélisation ontologique pour renforcer l'assistance aux acteurs en leur fournissant des outils d'inspection et de compréhension lors de la recherche. Cette assistance a été mise en œuvre dans un de nos prototypes, et évaluée empiriquement.

Table des matières

1 Ab initio **1**
- Domaine de recherche . . . 1
- Problématique . . . 2
- Contributions scientifiques . . . 4
- Plan du manuscrit . . . 6

I État de l'art 7

2 Comment analyser les traces d'apprentissage ? **11**
- Introduction . . . 11
- 2.1 Analyser des données éducatives . . . 11
- 2.2 Outils d'analyse . . . 18

3 Comment capitaliser les processus d'analyse de traces ? **25**
- Introduction . . . 25
- 3.1 Capitaliser au sein de la communauté EIAH . . . 25
- 3.2 Capitaliser hors de la communauté EIAH . . . 33

4 Comment structurer et utiliser l'information relative aux processus d'analyse ? **41**
- Introduction . . . 41
- 4.1 Modélisation sémantique . . . 42
- 4.2 Interpréter et exploiter la sémantique à travers les requêtes utilisateurs . . . 55

5 Synthèse **65**
- Introduction . . . 65
- 5.1 Classifications des propriétés des modèles et des outils d'analyse . . . 65
- 5.2 Verrous scientifiques . . . 72

II Contributions théoriques 75

6 Concepts fondamentaux de notre approche **79**
- Introduction . . . 79
- 6.1 Un cycle d'élaboration et d'exploitation des processus d'analyse issu de la littérature . . . 79
- 6.2 La capitalisation et ses propriétés intrinsèques . . . 86
- 6.3 Notre approche pour capitaliser . . . 88

7 Abstraire les processus d'analyse de traces **93**
- Introduction . . . 93
- 7.1 Description de l'approche . . . 94
- 7.2 Méthodologie de création des méta-modèles . . . 96
- 7.3 Formalisation du méta-modèle CAPTEN-ALLELE . . . 97
- 7.4 Illustration . . . 103
- Ce qu'il faut retenir . . . 105

8 Capitaliser les processus d'analyse de traces : de leur abstraction à leur narration **107**

Introduction . . . 108
8.1 Description de l'approche . . . 109
8.2 Méthodologie de construction de l'ontologie . . . 114
8.3 Notre framework ontologique CAPTEN-ONION pour la narration . . . 115
8.4 Illustration . . . 124
Ce qu'il faut retenir . . . 127

9 Assister la réutilisation *via* une recherche intelligente par inférence sémantique 129
Introduction . . . 129
9.1 L'importance d'un nouveau mécanisme de recherche . . . 130
9.2 Description de l'approche . . . 131
9.3 Formalisation de CAPTEN-FRUIT pour l'assistance à la recherche . . . 134
Ce qu'il faut retenir . . . 141

III Mises en œuvre 143

10 Indépendance technique des processus d'analyse 147
10.1 Présentation du prototype CAPTEN-APE . . . 147
10.2 Exemple d'utilisation de CAPTEN-APE . . . 149

11 Narration des processus d'analyse pour les rendre capitalisables 153
11.1 Introduction . . . 153
Introduction . . . 153
11.2 Présention du prototype CAPTEN-TORTOISE . . . 154
11.3 Peuplement du prototype avec l'existant . . . 159
11.4 Discussion . . . 161

12 Assistance *via* la recherche sémantique : CAPTEN-SEED 163
Introduction . . . 163
12.1 Présention du prototype CAPTEN-SEED . . . 164
12.2 Discussion . . . 167

IV Expérimentations et évaluations 169

13 Évaluation de l'abstraction des processus d'analyse de traces 173
Introduction . . . 173
13.1 Méthodologie d'évaluation . . . 173
13.2 Résultats expérimentaux . . . 175
13.3 Discussion . . . 177

14 Évaluation de la narration des processus d'analyse de traces 179
Introduction . . . 179
14.1 Méthodologie d'évaluation . . . 180
14.2 Résultats expérimentaux . . . 181
14.3 Discussion . . . 186

15 Évaluation de la recherche sémantique 189
Introduction . . . 189
15.1 Scénarios d'usage . . . 189
15.2 Discussion . . . 200

V *In Fine* 201

Conclusion 203

Perspectives **207**
Développer une banque commune de processus d'analyse narrés 207
Détecter les éléments critiques dans les processus d'analyse narrés 208
Adapter automatiquement les processus d'analyse de traces 208
Instancier automatiquement dans les outils d'analyse 209
Cycle de vie des processus d'analyse narrés exploitant la sémantique 210
Narration et science ouverte 210
Tendre vers un raisonnement à partir de cas 211

VI Bibliographie **215**

VII Annexes **231**

A Exemple de fiches techniques pour l'élaboration des opérateurs indépendants **233**
A.1 Fiche resumée d'un opérateur implémenté 234
A.2 Fiche d'identification d'un concept d'opération 235

B méta-modèle des processus d'analyse indépendant **236**

C Liste des processus d'analyse narrés **237**

D Documents liés à l'expérimentation des concepts de CAPTEN-ALLELE **239**
D.1 Protocole 240
D.2 Questionnaire 249
D.3 Grille d'évaluation 254

E Documents liés à l'expérimentation des concepts de CAPTEN-ONION **259**
E.1 Protocole 260
E.2 Questionnaire 277

F Compléments concernant la recherche avancée CAPTEN-FRUIT **287**
F.1 Patrons de recherche 287
F.2 Requêtes SPARQL 287

Ab initio

1

> *« Voilà mes œuvres ! Je ne les publie pas par vanité et je ne dis pas comme Horace : Exegi monumentum. »*
>
> **— Alphonse de Lamartine**
> (Préface des Œuvres complètes de Lamartine, 1860)

Sommaire

Section	**Domaine de recherche**	**1**
Section	**Problématique**	**2**
Section	**Contributions scientifiques**	**4**
Section	**Plan du manuscrit**	**6**

Domaine de recherche

Dans un contexte d'apprentissage médié par des Environnements Informatiques pour l'Apprentissage Humain (EIAH), les données produites par de tels environnements sont appelées traces d'apprentissage ou simplement traces – terme que nous adoptons dans le reste de ce manuscrit. Une trace d'apprentissage est communément définie comme un ensemble d'informations numériques qui sont produites d'une part *via* l'enregistrement des interactions réalisées par un apprenant dans le cadre d'une activté pédagogique, et d'autre part *via* l'interprétation que le système fait de ces interactions utilisateur (LAFLAQUIÈRE, 2009).

De par sa nature, une trace d'apprentissage *peut* donc véhiculer des informations signifiantes d'un point de vue pédagogique, comme l'état des connaissances d'un apprenant lors de la réalisation d'une activité, ou encore les phénomènes d'apprentissages existants au sein d'un EIAH. C'est la raison pour laquelle on cherche à extraire cette hypothétique information disséminée dans la trace, pour pouvoir l'utiliser à l'amélioration et à la compréhension des dispositifs d'apprentissage mis en place (DIMITRAKOPOULOU, 2004). L'analyse des traces d'apprentissage a pour but d'extraire cette information.

Cette thèse se situe dans le domaine de recherche des EIAH, et plus particulièrement dans les domaines de l'analyse des traces d'apprentissage que sont l'Educational Data Mining [1] (EDM) et surtout les Learning Analytics [2] (LA). Pour apprécier les particularités de l'EDM et des LA concernant l'analyse des traces d'apprentissage, il faut noter que les EIAH sont un champ pluridisciplinaire par essence : sciences de l'éducation, informatique, didactique, pyschologie cognitive – entre autres – s'associent pour tenter de favoriser l'apprentissage. Les communautés de recherche EDM et LA proposent ainsi des moyens d'analyser les traces produites par ces environnements, tout en tenant compte de cette singularité pluridisciplinaire intrinsèque aux EIAH, relative aux traces et aux acteurs.

Bien que ces deux domaines partagent un objectif commun qui les amène régulièrement à collaborer et à s'enrichir mutuellement, ils possèdent néanmoins des caractéristiques propres. On constate que l'EDM cherche à explorer les mégadonnées (big data) éducatives à travers des méthodes de fouilles automatiques ou semi-automatiques, afin de mieux comprendre les processus d'apprentissage qui surviennent dans les EIAH. L'EDM cherche également à développer de nouvelles méthodes tenant compte des spécificités de ces données (EDUCATIONAL DATA MINING SOCIETY, 2018 ; ROMERO et al.,

1. Fouille de données éducatives
2. Analytique de l'apprentissage

2010a). Quant aux LA, ils revendiquent une approche holistique et itérative, centrée sur l'analyse de traces. Les LA s'intéressent à la manière de collecter, de mesurer et d'analyser les traces relatives aux apprenants et aux contextes pédagogiques, mais aussi à la manière de visualiser les connaissances produites ainsi qu'aux personnes destinataires de l'analyse (*e.g.* enseignants) (R. S. J. D. Baker et al., 2004).

La différence principale entre ces deux communautés (Siemens et R. S. J. D. d. Baker, 2012) est la place de l'humain dans ces études des traces. Dans une approche de type fouille de données, adoptée par l'EDM, la place et le rôle des différents acteurs jouent un rôle secondaire, du fait du caractère automatisé ou semi-automatisé de l'approche. En revanche, les LA font intervenir les acteurs et la pluridisciplinarité de leurs compétences pour mieux comprendre les situations pédagogiques, ainsi que pour faire émerger de nouvelles pratiques d'analyse pertinentes par rapport à des situations pédagogiques définies.

Cette thèse se situe principalement dans le champ des Learning Analytics, ceci afin de tenir compte de cette pluridisciplinarité. Nous ancrons également notre travail dans une approche d'ingénierie des connaissances, et étudions ces analyses d'un point de vue informatique, en les considérant comme des artefacts modélisables et interprétables par la machine. Ces artefacts qui suscitent notre attention s'appelent des *processus d'analyse de traces d'apprentissage* et sont définis dans la littérature comme une succession d'opérations appliquées à des traces d'apprentissage (Mandran et al., 2015), représentant en réalité la manière dont sont mises en œuvre ces analyses au sein d'outils d'analyse de traces.

Problématique

Un processus d'analyse de traces, et à plus forte raison l'analyse que ce processus met en œuvre, a pour objectif de répondre à un besoin d'analyse, à partir de traces d'apprentissage issues d'un contexte pédagogique donné, comme l'utilisation d'un EIAH donné. Il s'agit ici d'un moyen important pour la prise de décision, qui a une conséquence à la fois pédagogique et éthique (Slade et Prinsloo, 2013 ; Greller et Drachsler, 2012). La conception et la mise en œuvre d'un processus d'analyse sont des tâches réalisées par des analystes, qui peuvent être par exemple des chercheurs, ou encore des statisticiens. Ces tâches de conception et de mise en œuvre sont bien souvent complexes et fastidieuses, notamment car le besoin d'analyse doit être correctement cerné et les traces, ainsi que les connaissances liées au domaine et à son contexte, doivent être correctement communiquées aux analystes.

Cependant, ces tâches de conception et de mise en œuvre pourraient être facilitées par la possibilité d'accéder aux analyses existantes et de rechercher celles répondant à une question similaire, dans le but de trouver des processus d'analyse déjà réalisés pouvant être réexploités. Cela permettrait alors aux analystes de pouvoir mettre directement en œuvre des analyses, en considérerant les analyses comme des outils d'aide à la conception, leur proposant des méthodologies, des points de comparaison pour limiter les biais scientifiques ou des manières d'interpréter les connaissances produites : autrement dit, tirer parti de l'existant.

Il faut noter que la mise en œuvre de ces analyses est en général étroitement liée aux contextes pédagogiques dans lesquels les traces analysées ont été récoltées, ce qui provoque actuellement des difficultés (voire l'impossibilité) de se resservir des processus d'analyse dans d'autres contextes pédagogiques (Chatti et al., 2012). En effet, les techniques utilisées peuvent ne plus convenir dans ces nouveaux contextes et ainsi produire des résultats erronés (Suthers et Rosen, 2011). De plus, même dans le cas d'un contexte pédagogique similaire, les spécificités techniques rencontrées, liées aux format des traces et au contexte d'implémentation (*e.g.* outils utilisés), sont telles qu'elles complexifient ces analyses, rendent très difficile leur réutilisation directe et rendent leur partage au sein de la communauté peu propice (Clow, 2013).

Pourtant, l'on constate que la communauté est régulièrement confrontée à des besoins d'analyse similaires, mais se déroulant dans des situations pédagogiques différentes. De ce fait, nous pensons que les conséquences qui découlent des difficultés à partager, comprendre et réutiliser un processus

d'analyse sont dommageables pour la communauté. Tout d'abord, cela a un impact sur l'effort d'analyse de la communauté, puisque pour un même besoin d'analyse mais décliné dans des outils différents, l'on redéfinit autant d'analyses – ce qui empêche la communauté de tirer profit de l'existant (Dyckhoff et al., 2012). Il est de plus dangereux de fonder la prise de décision sur des analyses réutilisées alors qu'elles ne sont que partiellement compréhensibles, puisque des biais décisionnels importants peuvent alors se manifester, jusqu'à impacter l'apprenant lui-même (Dringus, 2012). Enfin, l'aspect technique des analyses les rend difficilement compréhensibles pour les acteurs non analystes pouvant entraîner une perte d'information importante lors de l'analyse ou de sa documentation.

Partant de ces constats, notre problématique est la suivante :

Problématique

Comment peut-on rendre les processus d'analyse de traces d'apprentissage capitalisables au sein de la communauté des Learning Analytics ?

Dans le cadre de cette thèse, nous avons cherché un moyen pour tenir compte de l'implication de tous les acteurs de l'élaboration d'une analyse, et pour rendre accessibles les processus d'analyse existants à toute la communauté. En outre, nous avons cherché comment les processus d'analyse pouvaient être réutilisés dans d'autres contextes pédagogiques, tout en tenant compte des spécificités pédagogiques et des informations initiales de l'analyse. Nous avons ainsi décomposé notre problématique de recherche en trois questions de recherche, qui tiennent compte des propriétés nécessaires à une *capitalisation* des processus d'analyse de traces d'apprentissage telle que nous l'envisageons.

Le premier problème est que les processus d'analyse sont mis en œuvre dans des outils d'analyse divers et variés, qui sont dotés de spécificités techniques propres. Par exemple, pour une même opération implémentée dans deux outils d'analyse différents, le paramétrage peut être différent pour des raisons techniques. L'outil influence donc la conception des processus d'analyse. La question est de savoir comment un processus d'analyse réalisé dans un outil d'analyse peut être mis à disposition de toute la communauté, indépendamment de l'outil d'analyse utilisé pour le créer. D'un point de vue *processus d'analyse*, il s'agit de se demander s'il est possible de s'affranchir des contraintes techniques générées par les outils d'analyse sans perdre d'information. Notre première question de recherche est donc :

Question de recherche 1

Comment partager et combiner des processus d'analyse mis en œuvre dans différents outils d'analyse ?

Le deuxième problème pour envisager une capitalisation des processus d'analyse au sein de la communauté réside à la fois dans la possibilité de *comprendre* et d'*adapter* un processus d'analyse, pour pouvoir s'en resservir, alors qu'il est développé dans un contexte pédagogique précis. La première interrogation concerne la compréhension des processus d'analyse pour qu'ils puissent être appréhendés par des acteurs aux expertises différentes, et ainsi leur fournir un cadre pour l'élaboration et la réutilisation de l'analyse. Il s'agit de s'interroger sur comment intégrer l'information en lien avec les processus d'analyse et comment proposer différents niveaux de lecture de l'analyse, pour les rendre intrinsèquement intelligibles aux acteurs. La deuxième interrogation concerne la difficulté à adapter un processus d'analyse. Il s'agit dans un premier temps de s'interroger sur l'impact des contextes pédagogiques et des choix de conception sur l'analyse. Il s'agit ensuite d'envisager comment ces informations peuvent être représentées et structurées pour enrichir le processus d'analyse. Notre seconde question de recheche est ainsi la suivante :

Question de recherche 2

Comment permettre de réexploiter un processus d'analyse existant pour répondre à un autre besoin d'analyse ?

Enfin, le troisième problème réside dans les risques liés à la réutilisation des processus d'analyse. En effet, l'aspect pluridisciplinaire de la communauté oblige à envisager des objectifs et des pratiques différents selons les acteurs, notamment lors de la consultation et de l'utilisation des analyses existantes. Toutefois, le cadre technique actuel dans lequel les analyses peuvent être mises en œuvre ne permet pas de répondre aux différentes attentes de ces acteurs. La conséquence majeure de cela est la non réutilisation de l'existant, ce qui empêche une réelle capitalisation et dessert la mise en commun des efforts de la communauté. La question est ainsi de savoir quelles assistances sont actuellement à la disposition des différents acteurs et comment les repenser pour qu'elles s'intègrent à une démarche de capitalisation. Cela requiert également d'identifier les besoins réels de ces acteurs pour mettre en adéquation ces besoins et la manière de les assister dans leur tâche. Du point de vue informatique, cela consiste à se demander comment traiter une information structurée contenue dans les processus d'analyse d'un environnement permettant le *partage* et la *réutilisation* de ces processus d'analyse.

Question de recherche 3

Comment assister les différents acteurs lors de l'élaboration et de l'exploitation de processus d'analyse ?

Pour conclure, si l'on conçoit un enviromment permettant de répondre à nos deux premières questions de recherche, en permettant de décrire des processus capitablisables, ce n'est qu'en abordant la troisième question que la communauté aura les moyens de réellement tirer parti de l'existant. Nous avons donc pour objectif de rendre les processus d'analyse de traces d'apprentissage capitalisables en permettant à ces processus d'être techniquement indépendants des outils d'analyse, mais aussi d'être compréhensibles, partageables et adaptables pour permettre leur réexploitation. De plus, les modèles ainsi proposés doivent permettre à ces propriétés de s'établir comme un socle à divers mécanismes d'assistance visant à promouvoir l'utilisation de l'existant au sein de la communauté.

Contributions scientifiques

Pour répondre à la problématique identifiée, nous proposons en premier lieu un cycle d'élaboration et d'exploitation des processus d'analyse, afin d'en définir les différentes étapes et les différents rôles. Ce cycle est issu de l'étude des étapes observables dans la littérature et des rôles attribués aux acteurs impliqués dans le domaine des LA. En outre, nous définissons la capitalisation et ses propriétés, et le cycle permet d'ancrer cette capitalisation dans un cadre formalisé. Nous proposons de plus un ensemble de contributions théoriques et de prototypes de recherche regroupées au sein d'un environnement nommé **CAPTEN** (*Capitalization of Analysis Processes for Technology Enhanced learNing*). Ces contributions théoriques sont issues d'une démarche itérative de notre part, avec la volonté de prendre en compte les retours utilisateurs lors de nos expérimentations, afin que nous puissions proposer une solution selon une démarche de co-construction avec les acteurs de l'analyse. Une vue globale de notre environnement, ainsi que de sa nomenclature, est visible dans la Figure 1.1 et servira de point de repère dans ce manuscrit.

Pour répondre à la première question de recherche, relative au partage des processus d'analyse indépendamment des outils d'analyse, nous proposons l'approche **CAPTEN-MANTA**. Ce travail s'appuie sur des méta-modèles permettant de décrire, indépendamment des outils d'analyse, les opérations utilisées dans les processus d'analyse, les processus eux-mêmes et les traces utilisées, afin de s'affranchir des contraintes techniques occasionnées par ces outils. **CAPTEN-MANTA** permet donc d'adopter un formalisme commun aux processus d'analyse pour envisager leur partage. Ce méta-modèle a été mis en œuvre et évalué *via* notre prototype **CAPTEN-APE**.

Pour répondre à la deuxième question de recherche, nous proposons l'approche **CAPTEN-ATOM**, qui est un incrément itératif de **CAPTEN-MANTA** suite aux limites identifiées durant son évaluation. Cette approche propose une ontologie des processus d'analyse utilisée pour définir un framework[3] narratif des processus d'analyse. Cette approche permet d'introduire de manière structurée des

3. cadriciel

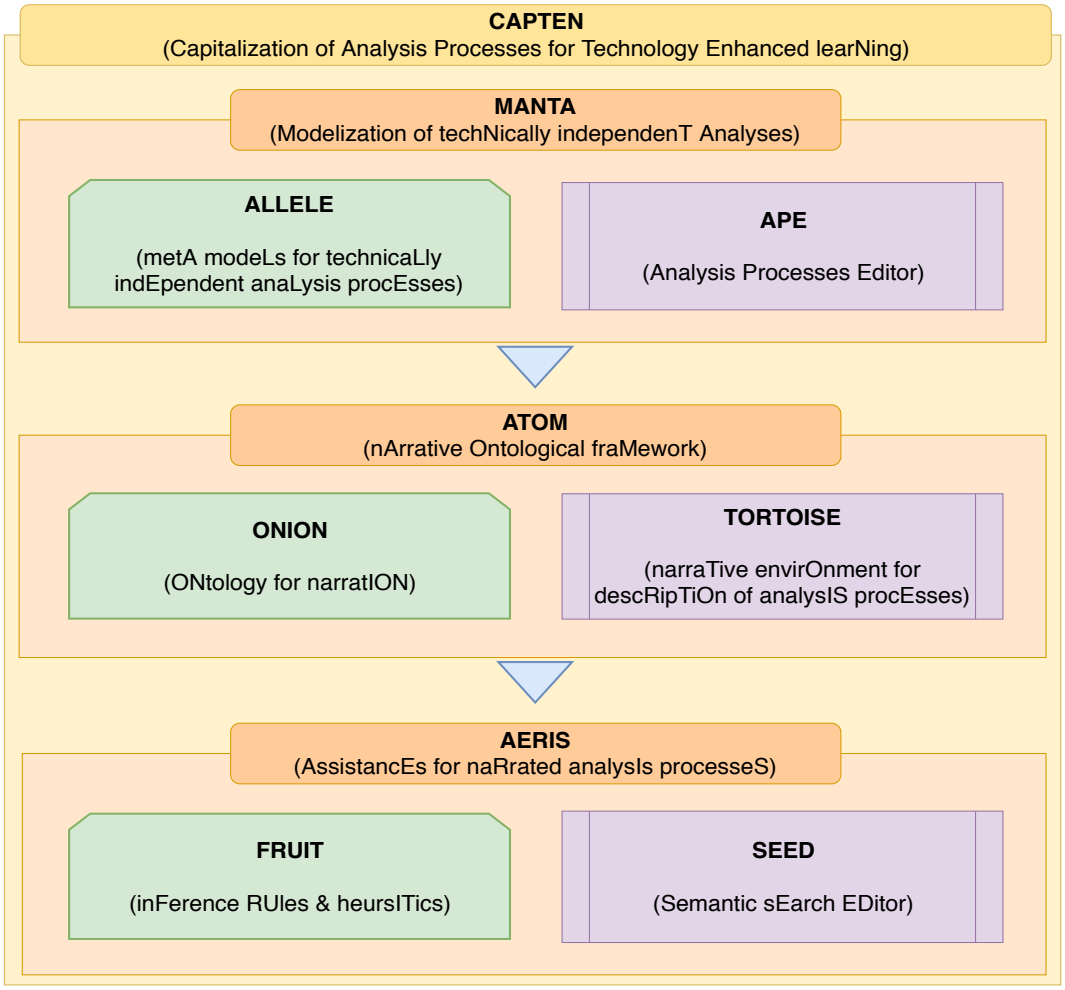


Figure 1.1.: Vue d'ensemble et nomenclature de notre proposition. Les rectangles à bords arrondis jaunes, oranges et rouges représentent nos différentes approches. Les hexagones irréguliers verts (semblable à une coupe de hangar) représentent nos contributions théoriques, et les rectanlges violets à bandes représentent la mise en œuvre de ces contributions.

éléments sémantiques dans la description des processus d'analyse. Cela nous permet d'enrichir le formalisme commun et indépendant des outils d'analyse proposé dans **CAPTEN-MANTA** et de satisfaire les propriétés nécessaires à la capitalisation. Ce framework a été mis en œuvre et évalué *via* notre prototype **CAPTEN-TORTOISE**.

Pour apporter des éléments de réponse à la troisième question de recherche, nous proposons **CAPTEN-AERIS**. Il s'agit d'une assistance à la recherche de processus d'analyse de traces d'apprentissage, d'après un besoin d'analyse spécifié par l'utilisateur. **CAPTEN-AERIS** exploite le cadre ontologique offert par **CAPTEN-ATOM** pour définir des règles d'inférences afin de raisonner sur les processus d'analyse dans leur ensemble. Cette approche permet donc d'assister "intelligemment" les différents acteurs de l'analyse dans l'exploration de processus répondant potentiellement à leur besoin d'analyse. Il permet également de fournir aux analystes des alternatives et des points de comparaisons issus d'autres processus lors de la phase d'élaboration de leur processus d'analyse. Cette assistance a été mise en œuvre dans notre prototype **CAPTEN-SEED**, et évaluée théoriquement.

Plan du manuscrit

La première partie de ce manuscrit de thèse est consacrée à l'état de l'art relatif à notre problématique de recherche. Dans le chapitre 2, nous étudions les analyses de traces d'apprentissage comme artefacts informatiques, la manière dont elles sont mises en œuvre, ainsi que les outils d'analyse utilisés dans la communauté. Dans le chapitre 3, nous approfondissons notre état de l'art en étudiant les efforts réalisés autour de la thématique de la capitalisation, tant dans notre domaine de recherche, que dans des disciplines différentes comme celle des workflows [4]. Enfin, telle que nous concevons la capitalisation, elle doit être un socle pour l'assistance aux différents acteurs, afin de réellement s'intégrer dans la pratique de la communauté. C'est pourquoi dans le chapitre 4 nous avons complété notre état de l'art en étudiant les différentes approches pour structurer et manipuler l'information en lien avec les analyses au sens large. Nous terminons cet état de l'art avec le chapitre 5 en présentant un comparatif des différents outils et techniques recensés, et les verrous actuels.

La seconde partie présente les différentes contributions théoriques que nous avons réalisées dans nos travaux, déclinée en quatre chapitres. Tout d'abord, dans le chapitre 6, nous proposons une définition de la capitalisation et de ses propriétés. Nous décrivons de plus un cycle d'élaboration et d'exploitation de l'analyse comme synthèse de l'état de l'art, que nous augmentons du rôle des acteurs, rendant patente la nécessité de capitaliser. Ensuite, dans le chapitre 7, nous proposons des méta-modèles permettant aux analystes de décrire les processus d'analyse existants de manière indépendante des outils d'analyse dont ils sont issus. Ces méta-modèles sont regroupés sous la dénomination **CAPTEN-ALLELE** et constitue la partie théorique de l'approche **CAPTEN-MANTA**. Par la suite, dans le chapitre 8, nous présentons **CAPTEN-ONION**, notre ontologie permettant la narration des processus d'analyse, qui constitue la partie théorique de **CAPTEN-ATOM**. Nous expliquons le formalisme et les principes utilisés pour couvrir les différentes propriétés nécessaires à la capitalisation, que nous illustrons sur un exemple. Enfin, nous présentons dans le chapitre 9 **CAPTEN-FRUIT**, le formalisme adopté, les régles d'inférence définies et les raisonnements sémantiques effectués dans **CAPTEN-AERIS** afin d'assister les différents utilisateurs lors de la recherche de processus d'analyse pour répondre à un besoin d'analyse.

La troisième partie est consacrée à la mise en œuvre de nos contributions théoriques *via* des prototypes de recherche. Pour chacune des approches, nous présentons le prototype associé, à savoir **CAPTEN-APE** pour l'approche concernant l'indépendance technique des processus d'analyse dans le chapitre 10, **CAPTEN-TORTOISE** pour leur narration dans le chapitre 11 et **CAPTEN-SEED** pour les mécanismes avancés de recherche de processus dans le chapitre 12. Pour chacun, nous expliquons leur fonctionnement, nous présentons ensuite leur architecture et situons nos propositions théoriques en leur sein. Enfin, nous montrons à travers des scénarios d'usage implémentables dans **CAPTEN-SEED** comment assister la recherche peut intéresser les différents profils d'acteurs de l'analyse.

La quatrième partie de ce manuscrit concerne les expérimentations menées pour évaluer nos trois contributions théoriques pour la capitalisation. Nous exposons les différentes évaluations menées ainsi que les résultats obtenus, que nous discutons également. Nous présentons en premier l'évaluation de notre approche **CAPTEN-MANTA** dans le chapitre 13, où les résultats ont contribué à penser la narration comme un candidat potentiel à la capitalisation. Ensuite, nous présentons dans le chapitre 14 les résultats expérimentaux de cette approche narrative, qui nous ont permis d'envisager les mécanismes avancés de recherche, que nous évaluons dans le chapitre 15.

Nous concluons cette thèse en présentant un bilan de notre travail. Puis, nous présentons différentes perspectives de nos travaux que nous jugeons captivantes pour la communauté des Learning Analytics, des EIAH, et aussi pour la science ouverte d'une manière générale.

4. flux de travaux

Partie I

État de l’art

Plan de la partie

Dans cette partie, nous étudions les travaux de recherche existants en lien avec notre problématique. Cette problématique concerne la définition d'une modélisation des processus d'analyse de traces d'apprentissage permettant leur capitalisation au sein de la communauté.

Notre problématique de capitalisation s'articule autour de deux axes principaux. Le premier axe concerne d'une part l'aspect technique des analyses, à savoir la manière dont les analyses sont mises en œuvre dans les outils d'analyse, les spécificités techniques de ces processus d'analyse ainsi que leurs propriétés computationnelles. D'autre part, cet axe traite aussi de la possibilité de partager et de réutiliser des analyses. Le deuxième axe concerne la façon de représenter et structurer les informations liées aux processus d'analyse, mais aussi la manière de les exploiter pour assister les différents acteurs lors de l'élaboration et de l'exploitation de tels processus.

Dans le chapitre 2, nous présentons les travaux en lien avec l'analyse des traces d'apprentissage. Nous nous intéressons tout d'abord aux manières de définir un processus d'analyse et à la finalité de ces différentes manières, avant de présenter les types d'outils d'analyse utilisés pour mettre en œuvre ces processus d'analyse, ainsi que certains de leur représentants, issus des travaux de diverses disciplines.

Dans le chapitre 3, nous présentons les différentes tentatives de partage et de réutilisation de la communauté, et observons comment une approche initialement centrée sur les données a peu à peu évoluée vers les processus d'analyse eux-mêmes. De là, nous explorons d'autres disciplines concernées par des besoins de partage, mais aussi de reproductibilité et de réutilisation.

Dans le chapitre 4, nous présentons les approches utilisées pour sémantiser et tenir compte des informations lors de l'analyse. Nous soulignons également les difficultés actuelles des outils d'analyse à exploiter ces informations pour permettre aux acteurs de rechercher des processus d'analyse. En conséquence, nous avons étudié des travaux en lien avec l'approximation des requêtes réalisées dans le domaine de l'ingénierie des connaissances et du web sémantique, ainsi que dans le domaine de la logique floue.

Nous concluons cet état de l'art en proposant une synthèse des différentes approches en lien avec la capitalisation, en adoptant une vision multi-domaine, pour souligner les lacunes existantes empêchant cette capitalisation. Enfin, nous indiquons les différents verrous à dépasser pour pouvoir répondre à notre problématique.

Comment analyser les traces d'apprentissage ?

2

Sommaire

Section	**Introduction**	**11**
Section 2.1	**Analyser des données éducatives**	**11**
2.1.1	Une histoire de traces d'apprentissage	11
2.1.2	Processus d'analyse de traces d'apprentissage : définition, objectifs, challenges	14
Section 2.2	**Outils d'analyse**	**18**

Introduction

Analyser les traces d'apprentissage consiste à en extraire des connaissances qui pourront être utilisées par différents acteurs (*e.g.* acteur pédagogique, chercheur), ceci afin de mieux comprendre et d'améliorer les situations d'apprentissage. Néanmoins, l'extraction de ces connaissances n'est pas spontanée et relève d'une démarche souvent itérative et complexe, dépendante d'éléments très différents. Ces éléments sont principalement les traces d'apprentissage – qui consituent la ressource primaire de n'importe quelle analyse – les opérations ainsi que les outils d'analyse – dans lesquels traces et opérations sont utilisées pour créer les processus d'analyse.

Dans ce chapitre, nous présentons comment les analyses de traces d'apprentissage sont actuellement réalisées au sein de notre communauté. Pour cela, nous définissons ce qu'est une trace, et montrons qu'il s'agit d'un objet complexe par nature. Nous présentons ensuite les processus d'analyse à travers la vision de la communauté et nous montrons les difficultés inhérentes à la capitalisation de ces processus. Nous concluons ce chapitre par une présentation de différents outils d'analyse, majoritairement disponibles au sein de la communauté. Nous y soulignons les spécificités de chacun de ces outils et leurs caractéristiques préjudiciables à la capitalisation au sein de la communauté.

2.1 Analyser des données éducatives

2.1.1 Une histoire de traces d'apprentissage

Préliminairement à l'étude des processus d'analyse de traces d'apprentissage, il convient d'étudier ces traces d'apprentissages, leur spécificité ainsi que la raison de leur exploitation.

Les traces numériques d'apprentissage sont produites par l'utilisation d'environnements informatiques lors de situations d'apprentissage. Ces environnements peuvent capturer les actions des apprenants avec différents degrés de granularité, allant d'une capture fine, comme une frappe clavier, aux interprétations du système, comme le diagnostic effectué par un tuteur intelligent. Néanmoins, il convient d'observer qu'il existe une grande diversité d'EIAH, chacun possédant des particularités : elles se manifestent jusque dans les traces d'apprentissage que ces environnements produisent, très diverses, tant sur leur format que sur leur contenu et leur granularité.

Les LMS (*Learning Managment Systems*), type le plus couramment utilisé dans le monde éducatif, permettent de diffuser du contenu pédagogique, le plus souvent pré-établi. Moodle est un exemple de LMS très répandu dans les universités. La littérature montre que ce type d'environnements amasse un

volume conséquent de traces variées concernant un grand nombre d'apprenants (Romero et al., 2008 ; Ferguson, 2012). Ces traces peuvent par exemple contenir les ressources pédagogiques consultées par l'apprenant (*e.g.* vidéo) les messages postés au sein d'un forum ou d'une salle de clavardage ou encore les connexions de l'apprenant.

Les ITS (Intelligent Tutoring Systems) sont un autre type d'EIAH. Ces derniers visent à fournir aux apprenants des instructions, des retours d'expérience et des parcours pédagogiques personnalisés, le plus souvent sans intervention des enseignants, avec des techniques automatiques. Pour ce faire, la granularité des traces dans ce type d'environnement doit être plus fine, comme le nombre de fois qu'un apprenant a cliqué sur un bouton d'aide, ou encore le nombre d'aller-retours entre cours et exercices. Ce type d'EIAH place l'analyse des apprenants au centre de son dispositif d'adaptation automatique, ce qui peut requérir du système des informations supplémentaires intégrées aux traces et propres à l'environnement (Nkambou et al., 2010).

D'une manière générale, la littérature montre que les traces récoltées dans de tels environnements peuvent être regroupées dans trois catégories. La première catégorie concerne les actions réalisées par les apprenants dans le ou les environnements d'apprentissage (Siemens, 2012 ; Vahdat et al., 2015), habituellement temporalisées. La seconde catégorie concerne les traces issues de la collecte d'information grâce à un questionnaire soumis à l'étudiant, afin d'évaluer ses connaissances. Il s'agit par exemple de retours d'utilisation après un exercice, ou encore des formulaires pour demander certaines informations à l'apprenant, comme son cursus scolaire (Liaw, 2008). Ce type d'information, combiné aux actions, est réguilièrement utilisé pour obtenir des profils d'apprenants (Bienkowski et al., 2014). La dernière catégorie concerne l'EIAH lui-même, ses ressources et les données additionnelles produites tenant compte de son état (Nkambou et al., 2010).

Nous considérons donc une trace comme un ensemble d'éléments et d'événements, généralement temporalisés et non-continus, survenant au sein d'un ou plusieurs environnements d'apprentissage. Les éléments représentent l'état de ces environnements, et les événements représentent les actions réalisées par un acteur donné (*e.g.* apprenant, enseignant, système) dans ces environnements, qui ont occasionné un changement d'état du ou des environnements en question. Éléments et événements forment ainsi un ensemble d'états-transitions centré acteur (principalement l'apprenant). De plus, des relations supplémentaires, comme des informations de dépendance, peuvent exister dans cet ensemble formant la trace (Champin et al., 2004 ; Lund et Mille, 2009 ; Advanced Distributed Learning, 2013).

Bienkowski & *al.* (Bienkowski et al., 2014) proposent une vue schématique du flux des traces d'apprentissage, instancié dans le cadre d'un environnement d'apprentissage adaptatif, présentée dans la Figure 2.1. Ici, les traces qui proviennent de deux sources, à savoir les actions des apprenants et leur profil, sont utilisées par un modèle prédictif afin de produire des connaissances qui seront utilisées par les différents acteurs pédagogiques (*e.g.* enseignants, administrateurs) et le système pour améliorer l'apprentissage. Les auteurs placent ainsi l'analyse des traces d'apprentissage (point 3 de la Figure 2.1) comme élément central dans l'amélioration du contenu pédagogique et des moyens de prise de décision (point 4 et 5), puisqu'elle permet de fournir aux différents acteurs des connaissances sur les situations d'apprentissage. Ce modèle prédictif peut être le résultat d'une analyse précédente, réalisée par un analyste (*e.g.* statisticien, chercheur, expert en fouille de données).

Bien que les traces soient au centre du processus de prise de décision, elles n'en sont pas moins des éléments complexes contraints par des spécificités techniques induites par les environnements eux-même. Il existe certains travaux dont l'objectif est de proposer une vision commune de ces traces d'apprentissage, pour permettre par exemple leur partage au sein de la communauté, en uniformisant les actions et les activités des apprenants. xAPI (Advanced Distributed Learning, 2013) est l'un de ces travaux, que nous étudierons plus en détail dans la Section 3.1.1. Malgré ces travaux, des spécificités techniques demeurent toujours.

En effet, la forme des traces est aujourd'hui très diverse, incluant des relations riches, comme des compositions, des dépendances (*e.g.* temporelles, structurelles) ou encore une hiérarchie multi-niveaux de l'information (R. S. J. D. Baker et Yacef, 2009 ; L. S. Settouti et al., 2006). De plus, la manière de représenter ces traces varie énormément, avec des formats différents (*e.g.* CSV (Network Working

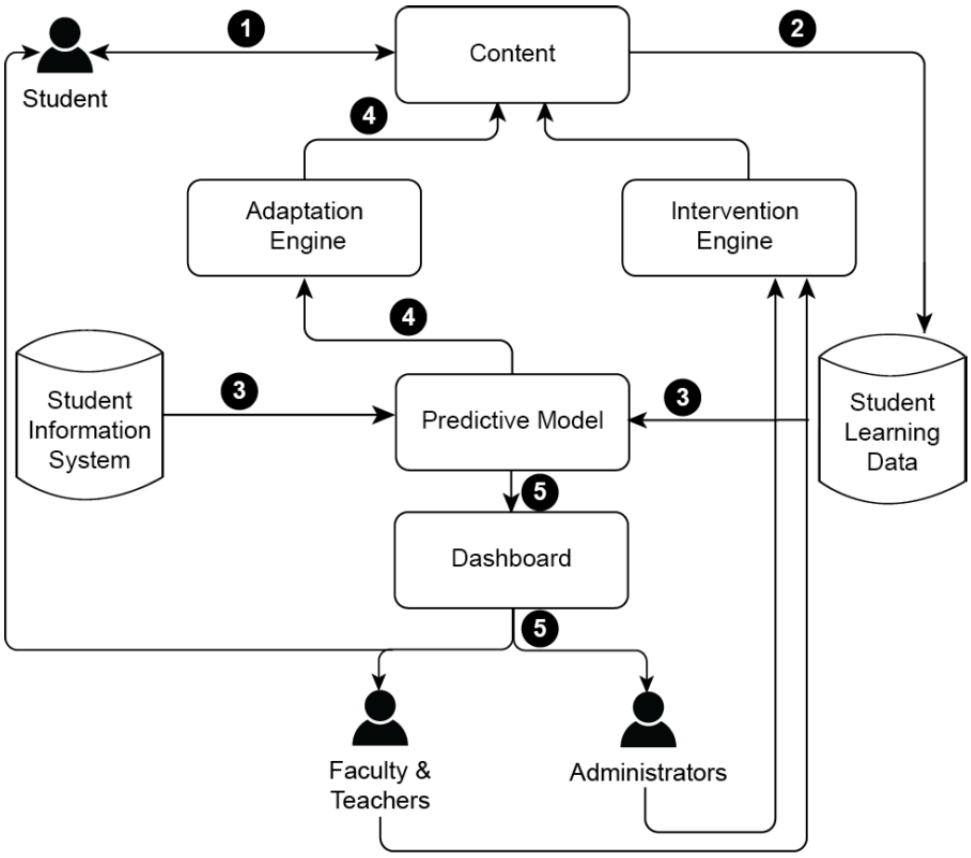


Figure 2.1.: Vue d'ensemble d'un flux de traces d'apprentissage dans le cadre d'un environnement d'apprentssage adaptatif, d'après (BIENKOWSKI et al., 2014, p. 18).

GROUP, 2005), JSON (DOUGLAS CROCKFORD, 2002), propriétaire de type Excel) qui peuvent avoir un impact sur la manière d'organiser l'information (LUKAROV et al., 2014) et, à terme, de la manipuler. Merceron et Yacef (MERCERON et YACEF, 2008) font d'ailleurs remarquer la difficulté à exploiter les fichiers de logs issus des LMS, notamment avec des techniques de fouille. En outre, ces traces peuvent aussi provenir de systèmes divers et contenir des informations fondamentalement différentes. La Figure 2.1 y fait écho : elle montre deux origines disctintes pour les traces *via* les traces issues du *Student Information System* et les *Student Learning Data*.

La Figure 2.2 et la Figure 2.3 illustrent cette diversité. La première figure est un extrait d'une trace anonymisée provenant de cours d'HarvardX et MITx dispensés sur la plateforme edX (MITX et HARVARDX, 2014). Il s'agit d'un fichier CSV où des informations liées à l'utilisation de l'environnement (*e.g. viewed*, *certified*, *ndays_act*) sont agrégées et référencées par des identifiants (*course_id* et *USERID_DI*). La seconde figure est un extrait d'une trace dans le format xAPI (ADVANCED DISTRIBUTED LEARNING, 2018[e]) décrivant une action réalisée par une personne au sein d'un EIAH. On remarque une différence significative de structuration et de sémantique entre ces deux formats de trace, comme la présence d'une hiérarchie de l'information dans la Figure 2.3.

Il en résulte alors une complexification de l'analyse, puisque les traitements mis en œuvre sur les traces doivent tenir compte de ces spécificités de représentation, et cela peut avoir une influence sur le processus d'analyse dans son ensemble. Un exemple intuitif est la définition de l'âge d'un apprenant dans une trace : cette variable peut être stockée sous la forme d'un entier dans une trace, et sous la forme d'une date (de naissance) dans une autre. Ainsi, si l'on veut pouvoir réutiliser un processus d'analyse qui, initialement, exploite l'âge sous forme d'entier, si l'âge est sous la forme d'une date, alors les traitements opérés dans cette analyse devront être adaptés, ou modifiés, afin de correspondre à cette spécificité.

course_id	userid_DI	registered	viewed	incomplete_flag	certified	LoE_DI	YoB	roles	gender	grade	start_time_DI	nevents	ndays_act
HarvardX/CB22x/2013_Spring	MHxPC130442623	1	0	1	0	NA	NA		NA	0	2012-12-19		9
HarvardX/CS50x/2012	MHxPC130442623	1	1	1	0	NA	NA		NA	0	2012-10-15		9
HarvardX/CB22x/2013_Spring	MHxPC130275857	1	0	1	0	NA	NA		NA	0	2013-02-08		16
HarvardX/CS50x/2012	MHxPC130275857	1	0	1	0	NA	NA		NA	0	2012-09-17		16
HarvardX/ER22x/2013_Spring	MHxPC130275857	1	0	1	0	NA	NA		NA	0	2012-12-19		16
HarvardX/PH207x/2012_Fall	MHxPC130275857	1	1		0	NA	NA		NA	0	2012-09-17	502	16
HarvardX/PH278x/2013_Spring	MHxPC130275857	1	0	1	0	NA	NA		NA	0	2013-02-08		16
HarvardX/ER22x/2013_Spring	MHxPC130198098	1	1		0	NA	NA		NA	0	2013-06-17	32	1
HarvardX/CB22x/2013_Spring	MHxPC130024894	1	1		0	NA	NA		NA	0.07	2013-01-24	175	9
HarvardX/CS50x/2012	MHxPC130024894	1	1	1	0	NA	NA		NA	0	2013-06-27		2

Figure 2.2.: Extrait d'une trace anonymisée, au format CSV, de cours d'HarvardX et MITx issus de la plateforme edX (MITx et HarvardX, 2014).

```
{
  "id": "http://example.com/activities/solo-hang-gliding",
  "definition": {
    "type": "http://adlnet.gov/expapi/activities/course",
    "name": {
      "en-US": "Solo Hang Gliding",
      "es": "Solo Ala Delta"
    },
    "description": {
      "en-US": "The 'Solo Hang Gliding' course provided by The Hang Glider's Club",
      "es": "El curso de 'Solo Ala Delta' siempre por el Club de Planeadores Hang"
    },
    "extensions": {
      "http://example.com/gliderClubId": "course-435"
    }
  }
}
```

Figure 2.3.: Extrait d'une trace décrivant un événement survenant au sein d'un EIAH, d'après (Advanced Distributed Learning, 2018[e]).

En conclusion, le format et le contenu des traces peuvent varier d'un EIAH à l'autre (*e.g.* jeu sérieux, MOOC), ainsi qu'en fonction de la matière qui y est enseignée (*e.g.* médecine, mathématique, informatique). Ces spécificités de format et de contenu rendent difficile la réutilisation des processus d'analyse pour des besoins similaires survenant dans d'autres contextes pédagogiques que celui initial (Papamitsiou et Economides, 2014 ; Dimitrakopoulou, 2004).

2.1.2 Processus d'analyse de traces d'apprentissage : définition, objectifs, challenges

Le but commun des domaines de l'EDM et des LA est, comme nous l'avons vu, de s'interroger sur la manière d'utiliser les traces d'apprentissage produites par les environnements afin de comprendre et d'améliorer le processus pédagogique dans son ensemble (Elias, 2011). Comme l'explique Baker (B. M. Baker, 2007), l'analyse des traces et son objectif se confondent dans le continuum des connaissances qu'il présente, illustré dans la Figure 2.4, où la finalité est de pouvoir utiliser des connaissances raffinées à partir d'éléments bruts pour répondre à des objectifs, ici, pédagogiques (*i.e.* besoin d'analyse).

Il est important d'étudier les besoins d'analyse auquels tentent de répondre l'EDM et les LA avant de s'intéresser à la manière dont sont produites les connaissances utilisées pour répondre à ces objectifs. En effet, ce sont les objectifs qui motivent la production de connaissances, et nous permettent ainsi

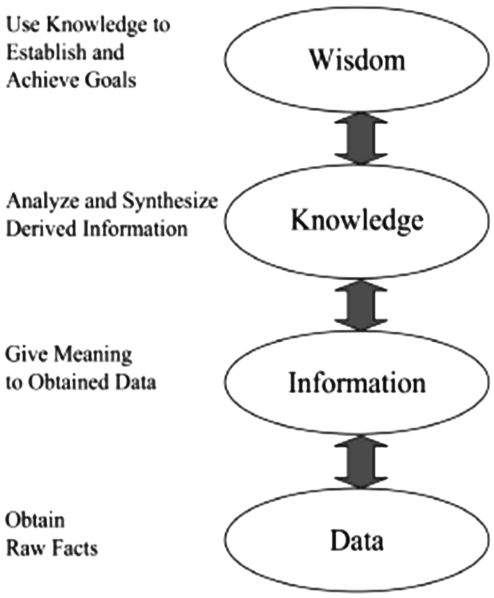


Figure 2.4.: Continuum des connaissances d'après Baker (B. M. Baker, 2007), récupéré dans (Elias, 2011, p. 6).

d'avoir une visibilité sur les méthodes d'analyse à adopter, les attentes du domaine, ainsi que les acteurs concernés (*e.g.* apprenants, enseignants) (Siemens et Long, 2011).

D'une manière générale, la littérature nous permet de définir cinq catégories de besoins qui sont étudiés (Papamitsiou et Economides, 2014 ; Nunn et al., 2016 ; Verbert et al., 2012) : (1) la prédiction de performance des apprenants, (2) la modélisation des connaissances de l'apprenant ainsi que de son comportement, (3) la procuration d'informations à l'apprenant et l'enseignant sur ses activités, (4) la prédiction du taux d'abandon ou de rétention, et (5) la recommandation de ressources.

Chacun de ces besoins est couramment associé à certaines méthodes d'analyse. Par exemple, la recommandation de ressources s'appuie majoritairement sur des méthodes de clustering et d'analyse séquentielle de motifs pour définir des recommandation personnalisées (Romero et al., 2009), alors que la prédiction de performance des apprenants a souvent recours à des analyses par régression, voire à des approches par réseaux neuronaux (Lykourentzou et al., 2009). Il faut cependant remarquer que dans la littérature, les analyses associées et les résultats obtenus ne sont que peu détaillés. Cela les rend difficilement reproductibles et applicables à d'autres contextes pédagogiques, en dépit du fait que les chercheurs jouent un rôle particulier dans la définition des bonnes pratiques d'analyse pour les différents acteurs (Nunn et al., 2016).

Ainsi, le coeur du problème pour les domaines de l'EDM et des LA est de parvenir à produire ces connaissances qui seront capables de répondre à des besoins d'analyse pour améliorer l'apprentissage, en analysant les traces d'apprentissage. Dans un contexte informatique, ces analyses sont mises en œuvre au sein d'outils et de programmes d'analyse et sont représentées par des processus d'analyse de traces d'apprentissage. Un extrait de processus d'analyse est visible dans la Figure 2.5. Ce processus d'analyse est implémenté avec la suite logicielle R (R Development Core Team, 2008) et se présente sous la forme d'instructions successives, mais d'autres formalisations existent, comme nous le verrons en Section 2.2.

Pour définir un processus d'analyse, nous reprenons les travaux de (Mandran et al., 2015) qui nous permettent de considérer un processus d'analyse comme une séquence non-linéaire d'opérateurs identifiables et réutilisables produisant *in fine* un résultat sensé, que nous appelons connaissance, et qui répond à des besoins pédagogiques (*e.g.* indicateur, modèle). Ces opérateurs peuvent être configurés selon certains paramètres et permettent de modifier l'état des traces sur lesquelles ils sont appliqués, ainsi que de créer de nouvelles données.

```
    cleaned_df = data_df[features_label]

    # Create a list of datapoints from the dataframe
    # Each element in the list will be a tuple with the form (label, features)
    # where label is either 1 or 0, and features is a list of the features from the dataframe
    clean_datapoints = map(list, cleaned_df.values)
    clean_labeledpoints = [(x[-1], x[:-1]) for x in clean_datapoints]

    Y_all = [x[0] for x in clean_labeledpoints]
    X_all = [x[1] for x in clean_labeledpoints]

    X_train, X_test, Y_train, Y_test = train_test_split(X_all, Y_all, test_size=.2, random_stat
e=0)

    # Create a logistic regression model from scikit learn
    logreg = linear_model.LogisticRegression()

    # Fit the logistic regression model using the training data
    logreg.fit(X_train, Y_train)
```

Figure 2.5.: Extrait d'un processus d'analyse implémenté avec R, réalisé par (AGNIHOTRI et al., 2016).

Cette défintion nous permet d'établir une vue plus précise de l'analyse des traces d'apprentissage. En s'autorisant à modifier la vue d'ensemble proposée par Bienkowski & *al.* (BIENKOWSKI et al., 2014) en y intégrant la définition de Mandran & *al.* (MANDRAN et al., 2015), nous schématisons ainsi, avec la Figure 2.6, la structure d'une analyse implémentée dans un contexte informatique par un analyste (*e.g.* statisticien, chercheur) : des traces d'apprentissage issues de sources hétérogènes sont manipulées dans un processus d'analyse afin de produire des connaissances exploitables par divers acteurs ainsi que par le système.

Cependant, l'obtention de ces connaissances n'est pas spontanée, puisque, d'après Baker (B. M. BAKER, 2007), ces connaissances résultent en réalité de l'analyse des éléments nommés "*Information*", eux-mêmes résultant d'éléments bruts nommés "*Data*" (cf. Figure 2.4) – assimilables aux traces d'apprentissage. Le distinguo entre ces deux termes (*i.e.* Informations et Data) est important, puisqu'il nous montre que les traces d'apprentissage ne constituent pas à elles seules une source d'informations suffisante, témoignant qu'elles ne sont pas assez porteuses de sens (ELIAS, 2011). Ces traces d'apprentissage doivent être enrichies par le biais de différentes méthodes pour pouvoir être comprises par l'analyste et à terme être considérées comme "informations" analysables pour produire des connaissances.

Cette étape d'enrichissement et de compréhension des données initiales est référencée dans la littérature comme le *prétraitement* des données. C'est une étape commune dans l'analyse des données de manière générale, qui consiste par exemple à nettoyer les données, à collecter les informations nécessaires à l'utilisation de certains outils ou encore à décider des stratégies de traitement des données manquantes (FAYYAD et al., 1996). Ainsi, les actions réalisées lors des prétraitements dépendent fortement des données concernées, du contexte technique de ces données et de l'objectif de l'analyse (VOLLE, 2001). De plus, les domaine de l'EDM et des LA doivent faire face à la spécificité du contexte pédagogique lors de ces étapes de prétraitement (R. S. J. D. BAKER et YACEF, 2009 ; ROMERO et VENTURA, 2010). Cette étape de prétraitement des traces est donc complexe et reconnue comme étant énergivore pour l'analyste. Dans le domaine de l'EDM par exemple, cette seule étape est estimée représenter entre soixante et quatre-vingt-dix pour cent de l'effort fourni par un analyste (ROMERO et al., 2014).

Le prétraitement des données précède donc l'analyse et lui est fondamental ; elle est recommandée par Romero & *al* (ROMERO et al., 2014) et fait partie intégrante de l'analyse et de son processus. Ce n'est seulement qu'une fois que les traces sont prétraitées – elles peuvent donc être considérées

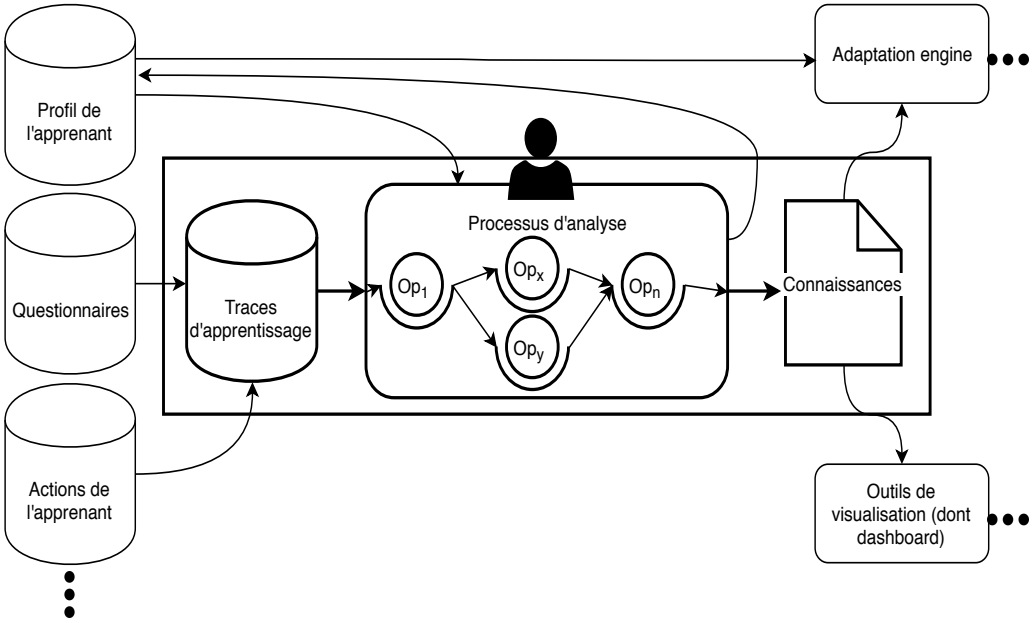


Figure 2.6.: Une vue d'ensemble de l'analyse de traces.

comme "*Information*" – qu'il est possible de les exploiter sans introduire de biais liés à une mauvaise interprétation (Huff et Geis, 1993).

Comme nous l'avons vu précedemment, il existe diverses méthodes pour analyser ces traces d'apprentissage (Papamitsiou et Economides, 2014 ; Nunn et al., 2016 ; Berland et al., 2014) : classification, clustering, régression, statistique, fouille par association de règles, celles dédiée à la fouille de textes, celles concernant analyse de réseaux sociaux, celles concernant la visualisation et enfin les méthodes de découverte avec les modèles (R. S. J. D. Baker et Yacef, 2009). Toutes ces méthodes sont mises en œuvre grâce à des opérateurs qui sont implémentés dans les outils d'analyse et ne s'excluent pas les unes les autres.

Parmi ces méthodes mises en avant par Baker & *al.* (R. S. J. D. Baker et Yacef, 2009), la découverte avec les modèles nous apporte des informations supplémentaires sur la manière dont considérer les processus d'analyse, leur structure et leur manière d'opérer avec les spécificités des traces d'apprentissage dans les domaines de l'EDM et des LA. Cette méthode consiste à réutiliser des modèles préalablement obtenus dans d'autres analyses pour répondre à de nouveaux besoins d'analyse. Par exemple, Jeong & *al.* (Jeong et Biswas, 2008) étudie l'impact des variations d'un ITS sur l'interaction des apprenants en définissant d'abord un modèle de Markov caché (HMM [1]) d'après les événements d'un EIAH, pour après le réutiliser afin d'identifier les comportements des apprenants.

Ce qu'il faut remarquer ici est que cet HMM est utilisé en tant qu'opérateur d'un autre processus d'analyse. Cela met en avant la propriété des processus d'analyse de traces d'apprentissage à potentiellement jouer le rôle d'opérateurs pour d'autres processus d'analyse et ainsi être réutilisés dans d'autres circonstances. Cette propriété nécessite donc que les processus d'analyse soient partageables et suffisament bien implémentés pour pouvoir être adaptés et réutilisés en fonction du besoin. Cependant, les processus d'analyse sont fortement dépendants du contexte technique de leur implémentation, comme les spécificités des traces utilisées (*e.g.* formalisme) ou des spécificités techniques inhérentes aux outils utilisés. Ainsi, ces contraintes techniques ont une influence directe sur comment les processus d'analyse sont mis en œuvre : ils sont limités et représentatifs d'un contexte technique particulier. Il s'ensuit que le partage des processus d'analyse peut s'avérer non pertinent (Clow, 2013), et que leur compréhension peut s'avérer plus complexe et technique.

1. Hidden Markov Model

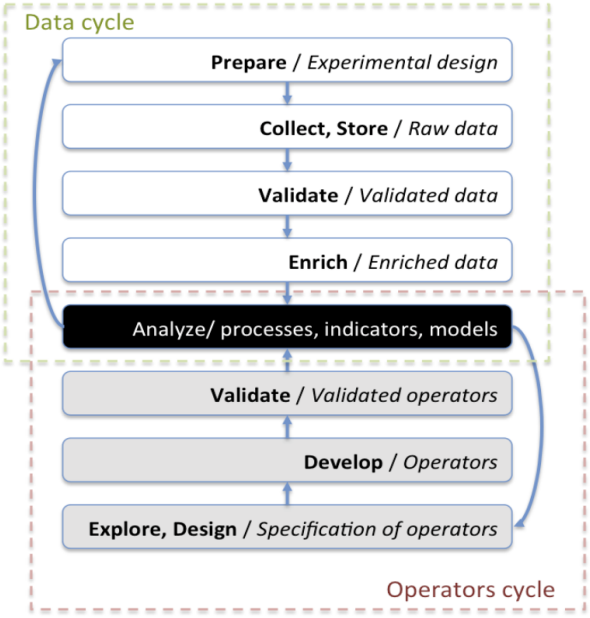


Figure 2.7.: DOP8 : un cycle de vie des opérateurs et des traces d'apprentissage d'après (MANDRAN et al., 2015, p. 2).

Finalement, cette proposition de méthode d'analyse avec les modèles vient corroborer notre constat que les processus d'analyse de traces sont de plus en plus considérés par la communauté comme une ressource à part entière, qui possède son propre cycle de vie et ses propres spécificités. Dans la littérature, on observe couramment que l'analyse est confondue dans le cycle de vie des données (J. C. STAMPER et al., 2011 ; FAYYAD et al., 1996). Ces dernières années, on observe que les opérateurs sont considérés comme des artefacts indépendants, évoluant de concert avec les traces d'apprentissage. Ainsi, la Figure 2.7 d'après Mandran & *al.* (MANDRAN et al., 2015) illustre DOP8, un cycle de vie des opérateurs d'analyse de traces, en retraçant les étapes de définition, de développement et de validation de ces derniers avant d'être utilisés sur des traces. Cela nous montre que la communauté commence à s'intéresser à la qualité de ces opérateurs, à leur relation avec les traces et à la manière dont ils sont implémentés dans les outils d'analyse, pour pouvoir être exploités.

2.2 Outils d'analyse

Les outils d'analyse sont des environnements dans lesquels sont menées les analyses et *a fortiori* dans lesquels sont mis en œuvre les processus d'analyse de traces d'apprentissage. Ce sont des environnements informatiques complexes bâtis sur la nécessité de calculer des données. Ces environnements adoptent des techniques différentes pour mettre en œuvre les processus d'analyse. Il disposent également de leurs propres opérateurs, capables de manipuler les données (MANDRAN et al., 2015), et imposent un formalisme de représentation des opérateurs et des données. Ainsi, à la fois les données et les opérateurs peuvent être très différents entre deux outils d'analyse, rendant difficile le partage des analyses et leur reproductibilité (SPRINGER NATURE, 2016). Par exemple, un opérateur de Classification Ascendante Hiérarchique (CAH) peut ne pas être implémenté de la même manière suivant les outils d'analyse – quand ceux-ci l'implémentent. Ainsi, la manière d'utiliser un tel opérateur et de le configurer peut changer entre les outils, comme c'est le cas entre Weka (WITTEN et al., 2016) et SPSS (NORUŠIS, 1990).

La Figure 2.5 (cf. section précédente) et la Figure 2.8 permettent de se rendre compte de cette diversité concernant la mise en œuvre des processus d'analyse. Alors que la première figure suggère une approche programmatique pour la mise en œuvre d'un processus dans R, la Figure 2.8 illustre une approche par workflow, réalisée dans RapidMiner (HOFMANN et KLINKENBERG, 2013). Cette seconde approche permet de représenter le flux d'opérations réalisées par le système lors du processus

d'analyse à un niveau d'abstraction plus important, au détriment d'une certaine flexibilité d'analyse. Le paramétrage d'un opérateur peut ainsi être moins complet qu'avec une solution programmatique comme R, ou bien encore l'ordre de la séquence d'exécution des opérations peut s'avérer peu ou non modifiable.

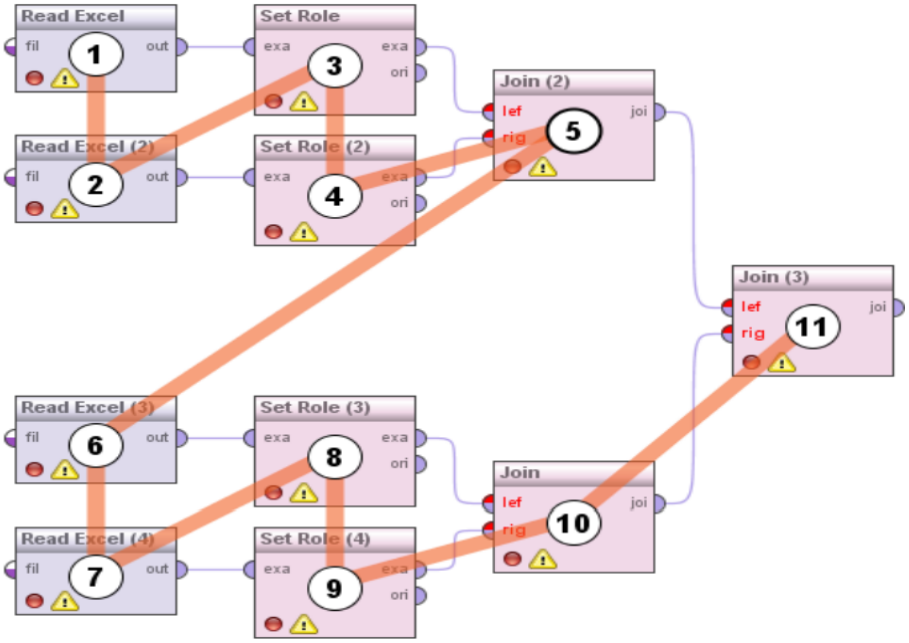


Figure 2.8.: Exemple d'un processus d'analyse sous forme de workflow, réalisé dans RapidMiner (RapidMiner, 2018[b], p. 43). L'ordre d'éxecution des tâches est indiquée par un nombre.

Grâce au caractère pluridisciplinaire des communautés EDM et LA, elles ont à leur disposition une grande diversité d'outils d'analyse, qui proviennent de diverses disciplines. On peut citer par exemple la suite logicielle R (R Development Core Team, 2008), Orange : Data Mining (Demšar et al., 2013) ou encore Weka (Witten et al., 2016). Ces trois outils sont issus respectivement des statistiques, de la fouille de données et de l'analyse de connaissance. D'une manière générale, l'avantage de réutiliser de tels outils est qu'ils ont déjà été éprouvés par leur communauté respective (Romero et al., 2010b).

Cependant, il apparaît évident que ces outils que l'on pourrait qualifier d'"inter-domaines" n'ont pas été développés pour tenir compte des spécificités inhérentes au domaine des EIAH et aux traces d'apprentissage. Cela se constate par le développement au sein de notre communauté d'alternatives et de compléments à ces outils d'analyse, qui proposent de nouvelles manières de représenter les traces et de les utiliser, comme nous le verrons dans les trois sous-sections suivantes. Verbert & *al.* (Verbert et al., 2012), suivis de Mandran & *al.* (Mandran et al., 2015), remarquent que ces initiatives font émerger une caractéristique forte du domaine : la communauté EIAH dispose de différentes catégories d'outils d'analyse.

En plus des outils dédiés à l'analyse comme Orange : Data Mining (Demšar et al., 2013) ou RapidMiner (Hofmann et Klinkenberg, 2013), qui sont considérés comme des outils dédiés aux spécialistes (Mandran et al., 2015), il est en effet possible de constater l'existence d'outils dédiés au stockage de traces, comme Mulce (Reffay et al., 2012) ou encore dataTEL (Verbert et al., 2011). Ces outils permettent de minimiser les étapes de prétraitement dans les analyses, en proposant des traces dans un format unifié : la mise en forme des données est faite en amont du processus d'analyse. En revanche, de telles spécifications de format contraignent les questions d'analyse auxquelles il est possible de répondre, puisqu'elles limitent par nature les informations pouvant y être représentées. Par exemple dans Mulce, le format permet notamment de répondre à des questions concernant l'amélioration des *social learning environments* (Reffay et al., 2012).

Enfin, une catégorie majoritairement propre au domaine des EIAH est celle des outils hybrides, combinant les deux catégories précédentes : le stockage et l'analyse. L'avantage de tels outils est de

pouvoir directement travailler avec des opérations capables d'utiliser les traces et les spécificités décrites dans un certain format, par exemple les actions réalisées par un apprenant sur une ressource précise. De plus, certains de ces outils permettent également de référencer (*e.g.* DataShop (KOEDINGER et al., 2010)) ou de stocker de nouveaux opérateurs pour ensuite pouvoir les réutiliser (*e.g.* UnderTracks (MANDRAN et al., 2015)). La conséquence directe est qu'il devient possible d'envisager un partage des processus d'analyse entre ces outils.

Cependant, comme remarqué précédemment, les spécificités des traces peuvent contraindre les questions qui peuvent être répondues. De plus, les opérateurs utilisés dans les outils d'analyse peuvent être difficiles à partager du fait de leur contexte technique (MANDRAN et al., 2015). Ainsi, bien que ces outils hybrides favorisent la réutilisation des analyses en interne, ils possèdent leurs propres spécificités techniques. Ces spécificités techniques influencent la manière dont sont implémentées les analyses, et rendent le partage des processus d'analyse complexe, contribuant de ce fait au manque d'interopérabilité des processus d'analyse au sein de la communauté (COOPER, 2013).

Néanmoins, ces approches hybrides constituent d'après nous des pistes solides pour tendre vers la capitalisation des processus d'analyse de traces d'apprentissage. Dans la suite de cet état de l'art, nous les présentons en détail.

Approche de type "workflows sur les données"

L'approche que nous qualifions de "workflows sur les données" consiste, pour un outil d'analyse, à disposer d'un entrepôt commun qui peut être directement requêté et utilisé à l'aide d'un système intégré d'analyse de haut niveau. Ce type d'approche permet une démarche exploratoire accrue lors de l'analyse, et offre aux analystes une sémantique plus importante concernant les opérations utilisées, tout en leur pemettant de considérer ces opérations comme des "boîtes noires". Un exemple d'une telle approche est présenté dans la Figure 2.9, qui illustre l'outil UnderTracks (BOUHINEAU et al., 2013b).

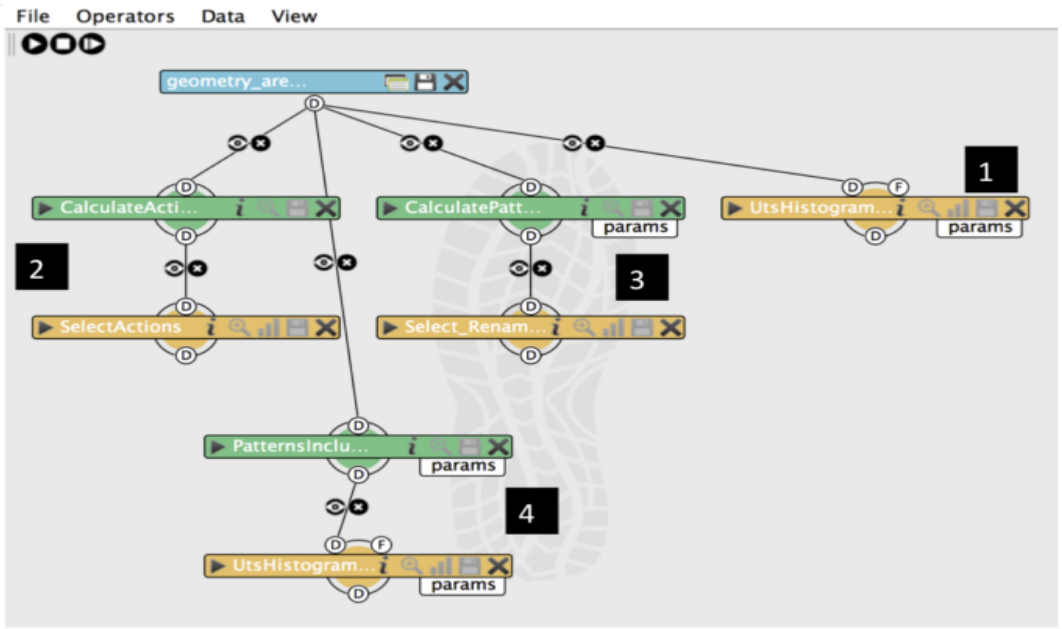


Figure 2.9.: Capture d'écran d'un processus d'analyse réalisé dans UnderTracks. Le rectangle bleu correspond au chargement des traces initiales, alors que les rectangles verts et jaunes sont des opérateurs, respectivement algorithmiques manipulant les traces et de visualisation (MANDRAN et al., 2015, p. 4).

Historiquement, DataShop peut être considéré comme le précurseur des outils d'analyse hybride (J. C. STAMPER et al., 2011 ; MANDRAN et al., 2015). Cet outil est principalement une plateforme de stockage de traces d'apprentissage dans un format de traces spécifique, en CSV (DATASHOP, 2010). Cependant cette plateforme permet également d'appliquer des opérateurs – bien qu'en nombre limité

car spécifique à un type d'EIAH (*i.e.* cognitive tutors) – pour manipuler directement les traces stockées. Cet outil fait parties du projet LearnSphere (LEARNSPHERE, 2018[b]), dont l'objectif est de munir la communauté d'outils pour l'analyse et le partage de traces. Récemment développé dans le cadre du projet LearnSphere, Tigris (LEARNSPHERE, 2018[d]) est un outil de création de workflows qui peut directement se brancher sur DataShop, et ainsi étendre les possibilités d'analyse de ce dernier.

Mandran & *al.* (MANDRAN et al., 2015) propose UnderTracks, une instanciation du cycle DOP8 qui définit un cycle de vie commun entre les traces et les opérateurs. Il s'agit d'un outil d'analyse permettant le partage de traces d'apprentissage et leur analyse, *via* une approche par workflows. Le format des traces est fondé sur une représentation sous forme de tables de bases de données, contenant les événements, les actions, les contextes et les apprenants, afin de permettre une description plus riche des événements pédagogiques.

Ce type d'approche a l'avantage de permettre de tester les analyses avec des données supplémentaires sans être confronté à des problématiques de formats ou d'outils, puisqu'elles sont mises en œuvre au sein d'un écosystème unique. Cet écosystème favorise également leur partage puisque le contexte technique y est identique. Cependant, nous avons constaté qu'il n'existe pas d'interopérabilité entre ces outils et leurs propriétés techniques sont très différentes. Par exemple, bien que Tigris et UnderTracks partagent une démarche similaire, Tigris est une solution JavaScript personnalisée, alors qu'UnderTracks est une extension de Orange : Data Mining, et dépend donc intrinsèquement de ses caractéristiques techniques.

Approche par "patron actionné par les senseurs"

Une autre approche consiste à utiliser un outil d'analyse capable de se greffer directement sur des environnements d'apprentissage, ou sur les traces produites, à l'aide de "senseurs". Ces senseurs permettent de définir une mise en correspondance entre les traces produites et celles nécessaires afin de rejouer une analyse préalablement mise en œuvre dans l'outil en question. Ce type d'approche met donc l'accent sur la réutilisation des processus d'analyse. Un travail représentatif de cette approche est IMS Caliper Analytics (IMS GLOBAL LEARNING CONSORTIUM, 2018[e]). Ce framework suggère de mettre en forme les données issues d'un système d'apprentissage à l'aide du senseur Caliper, sous la forme d'un triplet "Acteur-Action-Object" contextualisé, pour pouvoir ensuite appliquer des processus d'analyse sur ces traces homogénéisées.

Choquet & Iksal proposent également Usage Tracking Language (UTL), illustré dans la Figure 2.10, destiné à décrire l'obtention d'un observable à partir de données primaires issues de traces d'apprentissage (CHOQUET et IKSAL, 2006). La manière d'obtenir ces observables est définie dans la partie patron d'UTL (UTL/P). UTL/P permet d'indiquer les données sur lesquelles se fonde un indicateur (*i.e. derived datum*), elles-mêmes dérivées des données primaires. La partie UTL/T permet de mettre en lien les données produites par un environnement d'apprentissage avec les données primaires attendues, pour produire l'indicateur. La particularité de l'approche ici est de proposer un modèle dit de "Définition - Obtention - Utilisation" (*Defining, Getting, Using*, visible dans UTL/P), qui définit un ensemble de métadonnées explicatives de la donnée en fonction de son type (CHOQUET et al., 2009). Il est ainsi possible d'ajouter des informations structurées à ces données, possibilité que nous avons remarquée comme souvent absente des outils d'analyse de manière générale.

On constate que ce type d'approche est fortement centrée sur les données, et qu'elle semble considérer les propriétés de partage et de réutilisation des processus d'analyse comme découlant de ces mêmes propriétés pour les données. Or, l'approche par *workflows sur les données* vue précedemment nous a permis de remarquer que le partage et la réutilisation des processus ne dépendent pas que des traces disponibles, mais aussi des spécificités techniques. De plus, il n'existe pas d'intéropérabilité entre les analyses mises en œuvre dans ces différents travaux, ni entre les différentes approches présentées dans cette section.

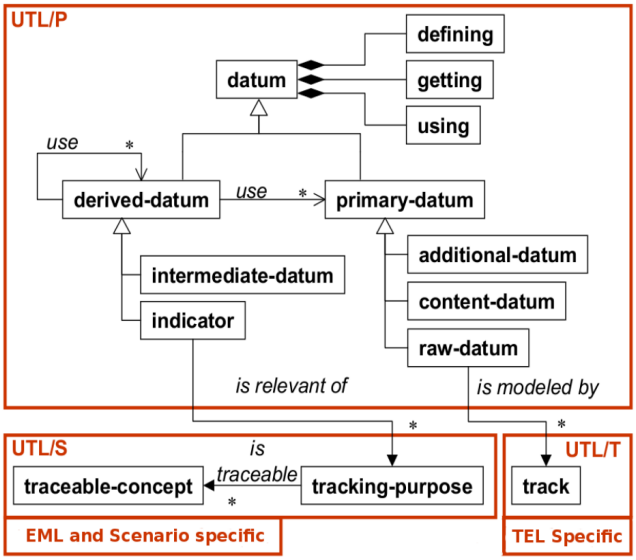


Figure 2.10.: Modèle conceptuel du langage UTL d'après (IKSAL, 2012, p. 88). UTL/P permet de décrire la structure d'un observable. UTL/S permet de lier la description des indicateurs à un scénario pédagogique. ULT/T permet de lier la description d'une trace à collecter avec les traces observées dans l'environnement d'apprentissage.

Approches tendant vers l'externalisation

Il est également possible d'observer d'autres approches d'analyse dans la littérature. Par exemple, Apereo Open Learning Analytics Platform (OLAP), qui est une initiative conjointe entre Apereo et SoLAR (APEREO FOUNDATION, 2016), se destine à être une approche "tout-en-un" dans le cadre des LA, en proposant des modules, certains encore en incubation, capables de collecter les traces, de les stocker, de les analyser et aussi de les communiquer et de proposer des mécanismes d'action en conséquence. Parmi les modules disponibles dans OLAP, le module Learning Analytics Processor (LAP) (APEREO FOUNDATION, 2018) se destine à mettre en œuvre les analyses. LAP est principalement conçu pour réaliser des analyses prédictives, avec pour objectif de pouvoir analyser de gros volumes de données. Pour ce faire, LAP modélise le processus d'analyse comme un ensemble de *pipeline* d'opérations prédictives ; un pipeline étant défini comme un ensemble "Entrée - Modèle prédictif - Sortie" configurable. Les modèles prédictifs appliqués aux traces peuvent quant à eux être définis en JAVA, *via* des bibliothèques JAVA annexes par exemple, ou appeler des services externes de calcul. Ce type d'approche par pipeline est intéressant car il permet d'isoler les étapes d'un processus et fait écho à l'approche par workflows.

On peut enfin citer l'approche *via* système à base de traces modélisées (SBT) (CHAMPIN et al., 2004 ; L. S. SETTOUTI et al., 2006), illustrée par la Figure 2.11. Un système à base de traces modélisées permet la collecte, le traitement des traces collectées et la visualisation de ces traces (LAFLAQUIÈRE, 2009). Ces traces sont considérées comme une suite d'éléments observés, appelés *obsels*, qui partagent des relations entre eux et sont *a minima* temporairement ordonnées pour tenir compte de la chronologie d'utilisation et/ou d'interaction observée lors d'une activité. Les traces stockées dans un SBT sont toujours accompagnées de leur modèle de traces, permettant de définir la sémantique des éléments qui s'y trouvent. L'extraction de connaissances s'effectue par des règles de transformation sur les traces et/ou les modèles de traces. Ces transformations peuvent d'ailleurs aboutir à de nouveaux modèles de traces, permettant de retracer l'évolution des traces.

Pour conclure, ces approches adoptent d'autres méthodes pour analyser les traces d'apprentissages et tenir compte de la spécificité du domaine, par exemple en proposant un modèle sémantisé de la trace. Cependant, du fait de leurs spécificités techniques particulières (*e.g.* modèle de traces spécifique pour le système à base de trace, bibliothèque optimisée pour le calcul prédictif, requêtage de services distants), les processus d'analyse sont confrontés à des contraintes techniques et conceptuelles nouvelles. Cela

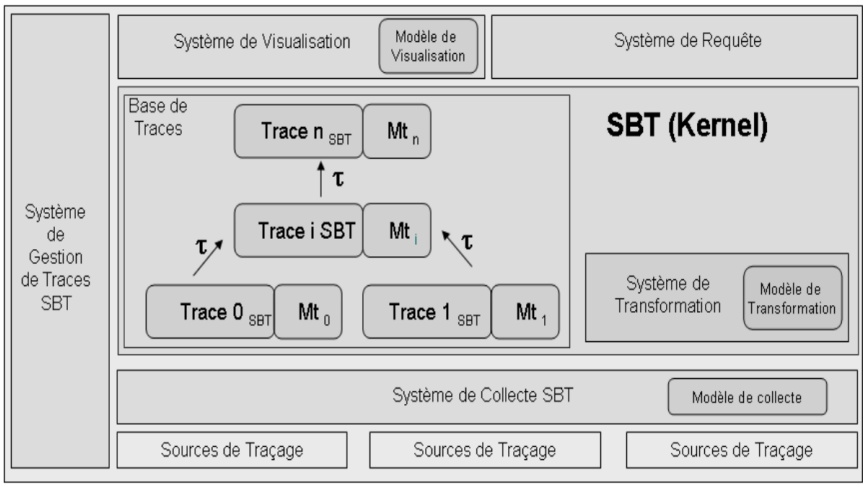


Figure 2.11.: Architecture du Système à Base de Traces, d'après (L. S. SETTOUTI et al., 2006, p. 5).

rend l'adaptation, la réutilisation et le partage de ces processus d'analyse complexes dans un autre contexte technique ou pour d'autres situations pédagogiques : il y a donc là encore un manque concernant leur intéropérabilité à travers la communauté.

Dans le chapitre suivant, nous étudions les travaux concernés par le besoin d'interopérabilité, de partage et de réutilisation des processus d'analyse et les propositions de ces travaux pour lever certaines contraintes techniques inhérentes aux outils d'analyse.

Comment capitaliser les processus d'analyse de traces ?

3

Sommaire

Section	**Introduction**	**25**
Section 3.1	**Capitaliser au sein de la communauté EIAH**	**25**
Section 3.2	**Capitaliser hors de la communauté EIAH**	**33**

Introduction

Capitaliser les processus d'analyse de traces d'apprentissage consiste à les mettre à disposition puis à pouvoir les réutiliser dans des contextes techniques différents et dans des contextes pédagogiques différents. Cela signifie que la chaîne des traîtements opérés sur les traces doit être partagée correctement, ainsi qu'adaptée aux variations apportées par ces changements contextuels. Néanmoins, comme nous l'avons vu précédemment, les processus d'analyse reposent grandement sur les traces, l'information qui s'y trouve, et comment elle y est représentée. Cette dépendance a grandement orienté la question de la capitalisation autour des traces et de leur partage, sans tenir compte des spécificités des analyses elles-mêmes.

Dans ce chapitre, nous présentons les travaux en lien avec la capitalisation. Nous commencons par ceux de notre communauté, en présentant comment la capitalisation par les traces est envisagée, et nous identifions les limites d'une telle approche. Nous présentons ensuite les travaux de la communauté concernant les processus d'analyse eux-mêmes, qui manifestent d'une prise de conscience de la communauté sur l'importance de les considérer comme des ressources à part entière, au même titre que les traces. Enfin, nous explorons des travaux issus d'autres communautés également concernées par la capitalisation des processus d'analyse, au sens général, et étudions leurs spécificités.

3.1 Capitaliser au sein de la communauté EIAH

Il n'existe pas, à notre connaissance, de travaux permettant de capitaliser les processus d'analyse de traces d'apprentissage au sein de la communauté. Néanmoins, dans cette section, nous présentons différents travaux pouvant partiellement répondre à cette problématique de la capitalisation, souvent de manière indirecte. Ces travaux concernent en premier lieu une approche très centrée données, et montrent comment la communauté a d'abord considéré la possibilité de pouvoir réutiliser les processus d'analyse uniquement grâce à des contenus pédagogiques harmonisés et des données formalisées. Puis, en second lieu, nous montrons comment cette vision a évolué vers le partage et la réutilisation des processus d'analyse de traces, grâce à des travaux plus récents. Nous étudions enfin les spécificités du contexte pédagogique qui agissent comme un frein à la capitalisation.

3.1.1 Des approches centrées données

D'une manière générale, les traces produites par les EIAH sont hétérogènes par nature et véhiculent des informations très différentes, comme des scores liés à une ressource spécifique ou une modélisation non standardisée du profil de l'apprenant (Desmarais et R. S. J. D. Baker, 2012). On constate alors que ces traces dépendent fortement de l'EIAH en question, de ses ressources et de la manière dont il a

été implémenté. Cette hétérogénéité pose des difficultés à la fois pour interpréter et pour échanger les traces (Desmarais et R. S. J. D. Baker, 2012 ; Bohl et al., 2002). Elle pose également des difficultés pour réutiliser les analyses existantes (Siemens et al., 2011).

Intuitivement, homogénéiser les traces produites par les EIAH semble alors être une solution pour permettre, grâce à leur partage, la réutilisation d'analyses déjà mises en œuvre. De nombreux travaux réalisés dans la communauté vont dans ce sens, et peuvent être regroupés en deux catégories.

La genèse de la première catégorie concerne la standardisation des contenus pédagogiques et des ressources à l'intérieur des systèmes d'apprentissage (Queirós et Leal, 2013). LOM (Learning Object Metadata), apparu dans les années 2000, est un exemple de ces travaux. Il permet d'indexer et de réutiliser des ressources pédagogiques dans différents environnements pédagogiques (IEEE, 2002). Ainsi, le contenus des traces et les ressources peuvent être représentés de la même manière, même si les environnements qui les produisent sont différents.

Progressivement, ce type de proposition a été enrichi par la notion d'activité des apprenants en lien avec ces ressources pédagogiques. C'est le cas de SCORM (Sharable Content Object Reference Metadata) (Advanced Distributed Learning, 2018[a]) par exemple, qui permet en plus de surveiller cette activité au sein de l'EIAH. Ces travaux montrent ainsi l'importance de considérer ces ressources comme des entités qui sont susceptibles d'évoluer au cours de la situation d'apprentissage et permettent une représentation homogène des actions sur les ressources pouvant se retrouver, à terme, dans les traces générées.

Cette notion d'activité réalisée par l'apprenant devient par ailleurs un nouveau paradigme pour la communauté, comme le montre, entre autres, l'émergence de modélisations comme EML (Educational Modeling Language) (Hummel et al., 2004) et IMS-LD [1] (IMS Global Learning Consortium, 2003). Ces travaux proposent la possibilité de scénariser l'activité pédagogique des apprenants dans un processus de formation (Pernin, 2004 ; Ferraris et al., 2005), en associant par exemple aux ressources des informations qui peuvent être comprises et traitées par la machine (IMS Global Learning Consortium, 2018[c]).

Bien que tous ces travaux ne concernent pas la trace directement, le fait d'homogénéiser la manière dont les contenus pédagogiques sont représentés dans les EIAH permet, *in fine*, d'envisager que ces contenus soient représentés de la même manière dans les traces. À terme, cela favorise donc le partage des traces. Néanmoins, il faut tout de même noter que les EIAH sont des environnements riches et complexes, et que de telles approches peuvent contraindre la diversité du contenu pédagogique envisageable, en plus d'être difficiles à mettre en œuvre.

La deuxième catégorie de travaux à propos du partage et de la réutilisation des traces concerne la formalisation des traces d'apprentissage et des informations qu'elles peuvent contenir. En effet, même si les contenus pédagogiques peuvent être formalisés, voire normalisés, la manière de produire les traces reste à la discrétion des concepteurs d'EIAH : en l'absence d'un format spécifique, ou en raison de contraintes techniques, une même action réalisée dans deux EIAH peut être représentée différemment. Cela crée des contraintes techniques supplémentaires et complexifie la réutilisation des traces, ainsi que les processus d'analyse qui en dépendent.

Dans le cadre d'une formalisation des traces, les traces peuvent soit être directement produites dans ce format, soit transformées *a posteriori* pour y correspondre. Cette deuxième méthode permet de conserver les spécificités des EIAH tout en bénéficiant de traces d'apprentissage standardisées, malgré un risque de perte d'informations si le format cible ne supporte pas toutes les informations présentes dans les traces. De cette manière, les traces d'apprentissage provenant de différentes sources peuvent être uniformisées et, dans le cadre de traces équivalentes, une même analyse peut être reproduite (Wagner et Ice, 2012).

Parmi les différents travaux de formalisation, nous pouvons citer tout d'abord ceux concernés par l'entreposage de traces d'apprentissage. L'objectif de ces entrepôts de traces est de centraliser et de rendre accessibles les traces d'apprentissage, qui proviennent généralement de sources différentes.

1. IMS-Learning Design

Pour pouvoir stocker de telles traces, ces entrepôts imposent en amont du dépôt un format précis, dans l'objectif de partager les traces et d'offrir un cadre propice à la reproduction et à la validation des analyses (Reffay et al., 2012 ; Koedinger et al., 2010). Dans DataShop par exemple (Koedinger et al., 2010), les traces d'apprentissage doivent être sous la forme d'un CSV et posséder des informations pouvant identifier l'apprenant, la session dans laquelle s'est déroulée l'action, le nom du problème ainsi que le moment auquel s'est déroulée l'action (Datashop, 2010).

Cependant, le format des traces choisi par ces plateformes de stockage ne répond pas toujours à un standard, puisque ces plateformes sont élaborées pour pouvoir répondre à des questions de recherche particulières (Mandran et al., 2015). Ainsi, on constate un manque d'intéropérabilité entre ces plateformes, dû aux formats spécifiques qu'elles utilisent (Duval, 2011).

D'autres travaux de formalisation des traces concernent la proposition de standards de traces d'apprentissage. Le propre de ces standards est de définir des formats cohérents et uniformes, capables de couvrir des besoins particuliers clairement définis. Néanmoins, ce type d'approche requiert que ces standards soient correctement répandus et largement utilisés, autrement ils s'apparentent eux aussi à des formats de traces quelconques.

Les deux standards les plus répandus au sein de la communauté sont actuellement xAPI (Advanced Distributed Learning, 2013) et IMS Caliper (IMS Global Learning Consortium, 2018[e]). Ces deux travaux s'inscrivent dans le paradigme qui a émergé *via* SCORM ou encore IMS-LD, au sujet de la notion d'actions des apprenants au sein de situations d'apprentissage. Une action suggère ainsi la présence d'une entité qui l'a réalisée, et l'objet sur lequel elle porte. xAPI et CALIPER formalisent donc *a minima* une trace avec, respectivement, un triplet $\langle Acteur - Verbe - Objet \rangle$ et un triplet $\langle Acteur - Action - Activité \rangle$ (IMS Global Learning Consortium, 2018[d]) : $Acteur$ représente l'entité réalisant l'action, $Verbe$ et $Action$ définissent l'action réalisée et $Objet$ et $Activité$ sont ce sur quoi porte l'action. L'avantage de ces standards est d'apporter un début de sémantique sur le type des informations contenues dans les traces.

Il est d'ailleurs important de noter qu'il existe une interopérabilité entre ces deux standards, favorisant de ce fait le partage des traces et la démocratisation de ces standards. De plus, ces deux travaux fournissent aussi des spécifications pour développer des entrepôts de stockage de traces dans leur format (les *Learning Record Store* pour xAPI et les *Event Store* pour Caliper), permettant ainsi de maintenir cette interopérabilité entre les entrepôts. En outre, ces travaux ne se limitent pas uniquement à la représentation d'une action sous forme de triplet : ils peuvent faire intervenir des éléments supplémentaires pour affiner la granularité de l'information (IMS Global Learning Consortium, 2018[b] ; Advanced Distributed Learning, 2018[c]). Certains de ces éléments ne se retrouvent pas dans les standards, ce qui peut impacter l'interopérabilité.

Par exemple, xAPI a recours à un champ d'extension supplémentaire lorsque le "type" d'une information ne correspond pas à un type préalablement défini (*e.g.* Acteur, Verbe), afin de ne pas la perdre. Ce champ d'extension laisse alors la modélisation de cette information entièrement à la discrétion des responsables des traces. Des traces avec une telle modélisation peuvent toujours être partagées dans le même standard xAPI, puisque le format est respecté, mais devront être prétraitées pour correspondre au format de CALIPER. Cela pose également des difficultés concernant la réexploitation de ces traces, puisque le contenu d'un tel champ d'extension n'est pas standardisé, ni sa sémantique (Advanced Distributed Learning, 2018[b]). L'interprétation des variables présentes et du contenu des traces peut varier en fonction des utilisateurs, mais aussi des contextes.

Ainsi, proposer une formalisation des traces permet de renforcer le partage et la réutilisation des traces d'apprentissage. Il s'agit d'une piste intéressante pour enrichir les traces d'apprentissage avec une sémantique sur le type des informations présentes. Puisque ces travaux permettent aussi de répondre à la problématique concernant la différence de représentation des traces, ils peuvent contribuer à diminuer la quantité d'étapes de prétraitement nécessaires lors des analyses. Néanmoins, ces travaux imposent par leur format des contraintes fortes sur l'information qu'il est possible de représenter et de convoyer. C'est pourquoi ces formats sont susceptibles d'orienter les questions de recherche et d'analyse auxquelles il est possible de répondre (Reffay et al., 2012 ; Mandran et al., 2015). Par ailleurs, ces travaux n'imposent pas de formalisme sur le *contenu* même des traces, ce qui limite la

possibilité de réutiliser les processus d'analyse avec d'autres traces, et n'apporte pas d'information sur comment adapter le processus en fonction de ces différences.

En conclusion, tous ces travaux ont permis d'affiner la manière dont il était possible d'envisager le processus d'apprentissage de l'apprenant. Ils ont surtout permis d'envisager une certaine interopérabilité entre les différentes traces d'EIAH, principalement grâce à des formats de traces communs et à des entrepôts de traces. Cela représente un réel atout pour la communauté, puisqu'il devient possible de tester les analyses sur un plus grand nombres de traces. Cependant, ces travaux sur les traces ne sont pas suffisants pour capitaliser les processus d'analyse de traces. En effet, ils n'apportent pas de solution sur la manière de partager et de reconduire les processus d'analyse définis dans des outils différents les uns des autres. Ils ne permettent pas non plus d'émanciper les processus d'analyse de leurs contraintes techniques, et n'apportent pas suffisament d'information sur le contexte pédagogique pour envisager d'adapter ces processus à des situations pédagogiques différentes.

3.1.2 Vers un partage et une réutilisation des processus d'analyse

À notre connaissance, les travaux concernant le partage et la réutilisation des processus d'analyse de traces d'apprentissage sont peu nombreux dans notre communauté, et récents. Ces travaux considèrent désormais les processus comme des ressources que l'on peut partager et réutiliser pour mener d'autres analyses (R. S. J. D. Baker et Yacef, 2009), et qu'il est nécessaire que ces processus soient accessibles à toute la communauté (Siemens et al., 2011).

L'un des premiers travaux que nous pouvons citer est UnderTracks (Bouhineau et al., 2013a). Comme nous l'avons vu dans la Section 2.2, il s'agit d'un outil d'analyse par workflows qui instancie le cyle de vie combiné des données et des opérateurs nommé DOP8. Cet outil stocke donc les traces d'apprentissage grâce à un entrepôt de traces dédié, accessible *via* une plateforme communautaire en ligne (Mandran et al., 2013). Le point intéressant de ce travail est qu'il fait également intervenir un entrepôt de stockage pour les opérateurs et pour les processus d'analyse de traces.

Classiquement, les outils d'analyse implémentent leurs propres opérateurs, comme Weka (Witten et al., 2016) ou SPSS (Norušis, 1990). Cette approche empêche de partager les opérateurs, c'est-à-dire les instructions à réaliser, avec d'autres outils d'analyse puisqu'ils sont implémentés dans le contexte technique de leur outil. Dans UnderTracks, les opérateurs disponibles à la (ré)utilisation proviennent de l'entrepôt avec lequel il est connecté. Cet entrepôt permet ainsi de centraliser les opérateurs et d'envisager leur partage entre différentes analyses, mais aussi entre différents outils. De plus, lorsqu'un nouvel opérateur est développé pour répondre à un besoin précis, il peut être déposé dans cet entrepôt et ainsi enrichir l'offre des opérateurs disponibles pour la communauté.

En outre, lorsqu'une analyse est mise en œuvre dans UnderTracks – en utilisant les opérateurs stockés, il est également possible de la déposer dans l'entrepôt des processus d'analyse : les opérateurs utilisés sont alors enregistrés, ainsi que leur configuration et leur séquence. Au même titre que les opérateurs, il devient ensuite possible de réutiliser dans UnderTracks les processus stockés par la communauté. Néanmoins, dans le cadre de leur réutilisation, la configuration de ces processus d'analyse devra être manuellement adaptée pour correspondre aux nouvelles traces à analyser. De ce fait, les processus stockés sont avec cette approche assimilables à des patrons d'analyse.

Ces entrepôts constituent un catalogue d'opérateurs et de processus d'analyse non négligeable pour la communauté, et offrent une solution exploratoire à la résolution des besoins d'analyse (Mandran et al., 2015). Supportés par la plateforme communautaire UnderTracks (Mandran et al., 2013), ces entrepôts sont ouverts à la communauté et peuvent être consultés en ligne. En revanche, l'on constate que l'outil d'analyse d'UnderTracks en soi ne permet pas de documenter les opérateurs et les processus d'analyse, contraignant ainsi leur compréhension. Pour pallier ce manque, il repose sur l'outil d'analyse Orange : Data Mining (Demšar et al., 2013) qui permet certaines descriptions (*e.g.* d'usage). De plus, la plateforme permet d'enrichir les opérateurs et les processus entreposés avec certaines métadonnées, comme l'objectif de l'analyse en question.

Un second outil qui permet le partage et la réutilisation des processus d'analyse est Tigris. Il s'agit d'un outil d'analyse par workflow en ligne, développé dans le cadre du projet LearnSphere (J. STAMPER et al., 2016 ; O'REILLY et HOFFMAN, 2017). Tigris utilise, entre autres, DataShop comme entrepôt de données, ce qui permet d'étendre les propriétés computationnelles natives de DataShop. À l'instar d'UnderTracks, Tigris permet de sauvegarder dans un entrepôt les processus d'analyse réalisés. Cet entrepôt est également ouvert à la communauté, et il est possible de parcourir les analyses mises en œuvre ainsi que de les réutiliser, là encore comme patron pour d'autres analyses.

En revanche, Tigris ne propose pas d'entrepôt d'opérateurs consultables et enrichissables, comme il est possible de le voir avec UnderTracks. À la place, les opérateurs sont référencés dans un dépôt Git (LEARNSPHERE, 2018[c]) et gérés par les propriétaires dudit dépôt. De plus, lors du workshop de la conférence Learning Analytics and Knowledge (LAK) 2018 concernant LearnSphere (LEARNSPHERE, 2018a), nous avons constaté que Tigris ne permettait pas non plus d'enrichir les processus d'analyse de traces avec des métadonnées supplémentaires (*e.g.* l'objectif de l'analyse), rendant la compréhension des analyses disponibles plus difficile.

Ces deux outils représentent une avancée majeure pour la communauté, car ce sont, à notre connaissance, les seuls à proposer à toute la communauté un entrepôt des processus d'analyse accessibles et exécutables. Cependant, bien que ces deux outils permettent le partage des processus d'analyse et leur réutilisation, il faut tout de même noter plusieurs limitations à ces travaux qui empêchent une réelle capitalisation des processus d'analyse.

Premièrement, ces deux outils possèdent leurs propres propriétés computationnelles, comme nous l'avons vu en Section 2.2. Par exemple, les opérateurs implémentés dans Tigris sont constitués de deux parties (LEARNSPHERE, 2018[e]) : une partie pour décrire la structure en XML Schema Definition (XSD), et un *wrapper* en JAVA pour encapsuler le comportement du programme. Toutes les spécificités de ces outils génèrent des contraintes techniques importantes et influencent la mise en œuvre des analyses. Ainsi, il n'existe pas d'intéropérabilité entre ces outils. Il n'est pas possible de partager les processus d'analyse d'un outil à l'autre pour qu'ils soient réutilisés, ni les opérateurs.

Deuxièmement, le manque d'une documentation associée aux processus d'analyse rend, au sein d'un même outil, l'intérêt de les partager et de les réutiliser limité. Actuellement, ces travaux ne permettent pas, ou peu, d'attacher de l'information supplémentaire aux processus d'analyse. Or, des travaux montrent que des informations minimales sont nécessaires pour espérer pouvoir comprendre les analyses et être en mesure d'interpréter correctement leurs résultats, notamment lors de leur reproduction ou de leur réutilisation (BÁNÁTI et al., 2015).

Pour cela, le projet ANR HUBBLE (PROJET HUBBLE, 2016) – dont cette thèse est issue – propose une plateforme destinée à stocker la description des analyses sous forme de texte libre (PROJET HUBBLE, 2018). Différents aspects y sont ainsi sauvegardés, comme les problématiques auxquelles répondent les analyses, certains éléments appartenant au contexte pédagogique duquel elles sont issues, ou encore une description des traces utilisées ou du processus d'analyse réalisé. Cela permet ainsi de proposer une vue d'ensemble de la mise en œuvre de l'analyse, à défaut d'un partage direct des processus d'analyse.

Néanmoins, dans tous ces travaux, les informations s'avèrent être faiblement structurées, à cause de l'utilisation d'annotations en texte libre. Or, comme nous l'avons vu, cela rend l'interprétabilité de ces informations complexe pour la machine.

Enfin, il faut noter que ces travaux ne s'intéressent pas à comment des analyses déjà mises en œuvre peuvent être appliquées à d'autres besoins d'analyses et à de nouvelles traces. En effet, ils ne permettent pas d'enrichir les processus d'analyse avec des informations en lien avec le contexte technique, comme les choix d'analyse (*e.g.* paramétrage) ou les ressources utilisées. Ils ne permettent pas non plus, lors du partage, d'indiquer si les processus fonctionnent correctement, ni d'indiquer la qualité des résultats obtenus. Ce manque d'informations entraîne, avec le temps, une dégradation des processus entreposés (BELHAJJAME et al., 2012b), en plus de causer une adaptation difficile de ces processus lors de situations nouvelles. Par exemple, on peut citer le fait que des services externes nécessaires à l'analyse ne sont plus disponibles, ce qui empêche de réutiliser le processus sans le modifier ; et s'il

n'existe pas d'informations en lien avec l'objectif et l'utilité de ces opérations distantes inaccessibles, il sera d'autant plus impossible de réutiliser le processus en question.

Parmi les travaux allant vers une capitalisation des processus d'analyse, nous pouvons également citer Usage Tracking Language (UTL) (Choquet et Iksal, 2007). Comme nous l'avons vu, UTL propose le modèle $\langle Définition - Obtention - Utilisation \rangle$, ou DGU, pour structurer l'information et la méthodologie relative à l'obtention d'un élément (*e.g.* indicateur, donnée primaire). Ce modèle a la particularité de s'adapter aux éléments qu'il décrit en modifiant les informations disponibles. La Figure 3.1 est un exemple des informations disponibles dans ce modèle pour une donnée intermédiaire. Une donnée intermédiaire est une donnée pouvant être issue à la fois de la manipulation de données primaires et d'autres données intermédiaires. Autrement dit, il s'agit du résultat d'un opérateur qui a été appliqué sur des données.

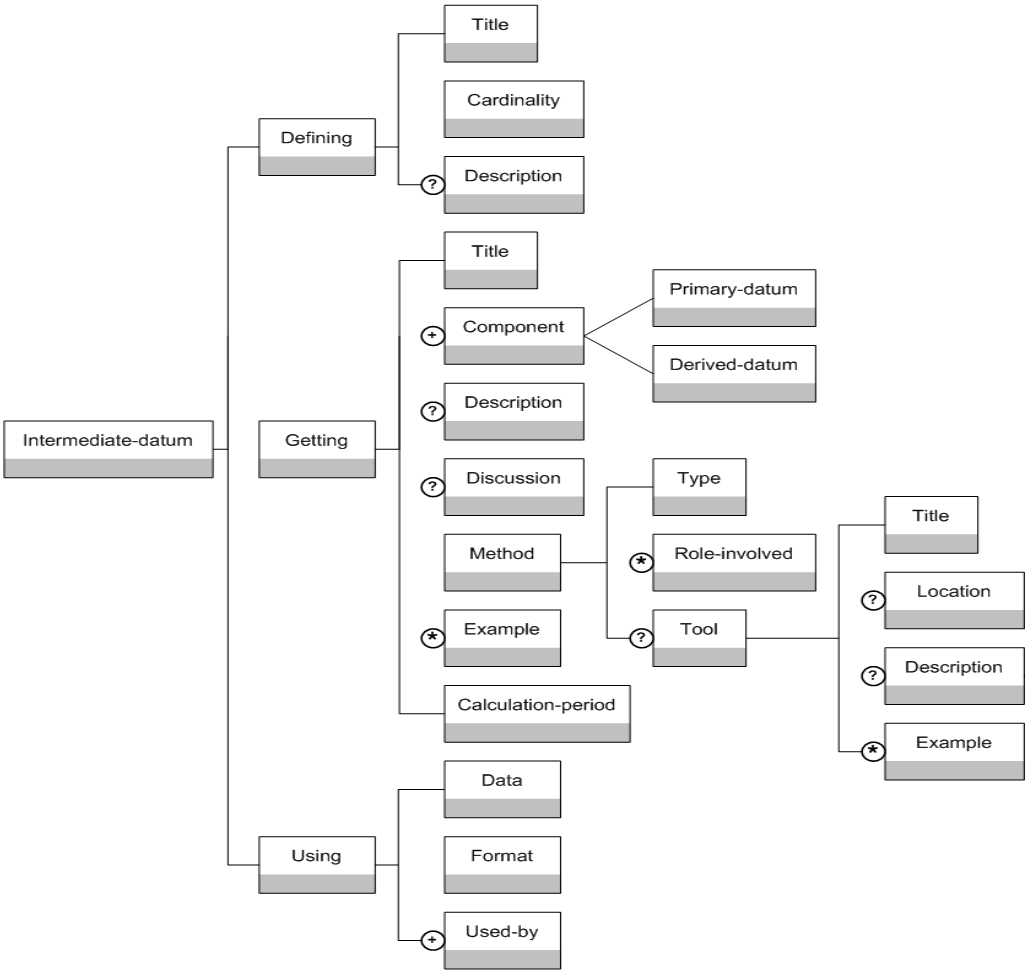


Figure 3.1.: Vue d'ensemble simplifiée du modèle $\langle Définition - Obtention - Utilisation \rangle$ (DGU) d'UTL (Choquet et Iksal, 2007, p. 15).

Au delà de la structuration de l'information qu'il permet, ce travail apporte également des propositions intéressantes concernant la réutilisation des processus d'analyse, mais aussi leur adaptation. La partie Utilisation du modèle (*Using* dans la Figure 3.1) permet d'indiquer certaines spécificités techniques de la donnée, comme son format, contribuant à renforcer les explications sur les choix d'implémentation. La partie Obtention (*Getting* dans la Figure 3.1) permet de caractériser la construction de la donnée : quelles données (*Component*) ont été utilisées, et avec quelles méthodes d'analyses (*Method*). La partie *Discussion* permet d'expliquer différents points sur l'obtention de la donnée, comme les choix réalisés ou les particularités qui peuvent exister. Il est de plus possible de fournir des exemples pour aider à la compréhension. Tous ces éléments du modèle DGU sont autant de supports supplémentaires pour aider à la compréhension de l'analyse, et favoriser son adapation à de nouvelles situations. Comme le

fait remarquer Stamper & *al.*, les métadonnées favorisent la réutilisation d'analyses déjà existantes, principalement parce qu'elles permettent de donner du sens aux traces utilisées (J. C. Stamper et al., 2011).

Pour finir cette section, il convient également de citer la manière dont les analyses sont représentées dans le Learning Analytics Processor (LAP) de la plateforme OLAP (Apereo Foundation, 2016). Bien que l'écosystème présenté ne semble pas pourvu d'un entrepôt dédié au stockage des opérateurs ni des processus d'analyse, le LAP adopte une approche axée sur l'intéropérabilité des processus d'analyse. Il se propose de modéliser les processus d'analyse d'après le standard Predictive Model Markup Language (PMML) (Data Mining Group, 2018[j]), dont nous parlerons dans la Section 3.2. Ce modèle permet principalement l'échange de modèles prédictifs entre différents outils. Grâce à lui, le LAP peut réutiliser des modèles prédictifs créés à partir de, et dans, d'autres outils d'analyse.

Néanmoins, la capitalisation des processus d'analyse ne dépend pas que d'un contexte technique, malgré l'orientation des travaux présentés dans cette section. Comme il est possible de l'entrevoir avec le modèle DGU d'UTL, le contexte pédagogique sous-jacent aux traces s'exprime également lors de l'analyse, et doit être pris en compte pour permettre d'adapter et de réutiliser les processus d'analyse.

3.1.3 La place du contexte dans la capitalisation : une contrainte supplémentaire et effective

Les travaux sur la formalisation des traces et des contenus pédagogiques permettent de nous rendre compte que les traces d'apprentissage sont par nature complexes. Elles proviennent d'un contexte pédagogique spécifique et potentiellement riche (*e.g.* ressources disponibles, moyens d'interactions disponibles). Dans cette section, nous montrons que les travaux actuels se heurtent à cette particularité du domaine lorsqu'il s'agit d'adapter et de réutiliser les processus d'analyse.

Si l'on s'intéresse plus en détail au contenu des traces d'apprentissage, l'on se rend compte qu'elles véhiculent des informations directement en lien avec le contexte pédagogique de la situation d'apprentissage dont elles proviennent (L. Settouti et al., 2010 ; L. Settouti et al., 2011). Ainsi, au sein des traces cohabitent des spécificités techniques et pédagogiques. Verbert & *al.* proposent une vue d'ensemble des différents types d'informations communément contenus dans les traces d'apprentissage (Verbert et al., 2012). La Figure 3.2 illustre cette vue d'ensemble, nommée *Learner Action Model*.

Bien que ce modèle puisse encore être enrichi, comme l'indiquent les auteurs, il permet de se rendre compte de la richesse des traces. Ce modèle met clairement en avant la présence du contexte pédagogique dans les traces, qui se matérialise selon différentes catégories (*i.e. learner*, *teacher*, *contexte*, *resource*, *action*, *type* et *result*), et avec une granularité différente (*i.e.* les clades de chaque catégorie). Il permet également de noter que certaines informations ne peuvent être propres qu'à certaines situations d'apprentissage. C'est par exemple le cas lorsque l'intérêt d'un apprenant pour une tâche est tracé, ou encore l'effet que peut avoir une situation sur l'apprenant.

À l'échelle de l'analyse, ce contexte pédagogique rentre en jeu et est crucial pour obtenir des résultats cohérents, ainsi que pour pouvoir comprendre et expliquer les résultats et les modèles obtenus (Vahdat, 2017). Il conditionne ainsi en partie les choix d'implémentation des processus d'analyse (*e.g.* paramétrages).

Pour tenir compte de ces spécificités pédagogiques lors des analyses et pouvoir les réutiliser, Chatti & *al.* (Chatti et al., 2012) suggèrent un modèle de référence pour les LA. Ce modèle, illustré en Figure 3.3, propose de conduire les analyses en suivant les quatre dimensions *Quoi*, *Pourquoi*, *Qui/Pour qui* et *Comment*, et de tenir compte des informations associées. Chacune de ces dimensions étant, de plus, associée, par l'entremise des boîtes rectangulaires, à des challenges et des questions de recherche.

Dans ce modèle, les techniques utilisées lors de la mise en œuvre des analyses, ainsi que les processus d'analyse, se situent dans la dimension "*Comment*" (quart inférieur gauche de la Figure 3.3). Cette dimension concerne des questions comme celle du design du processus d'analyse, de son utilisabilité, de ses performances ou encore de sa possibilité à être étendu. Bien qu'il soit possible de remarquer

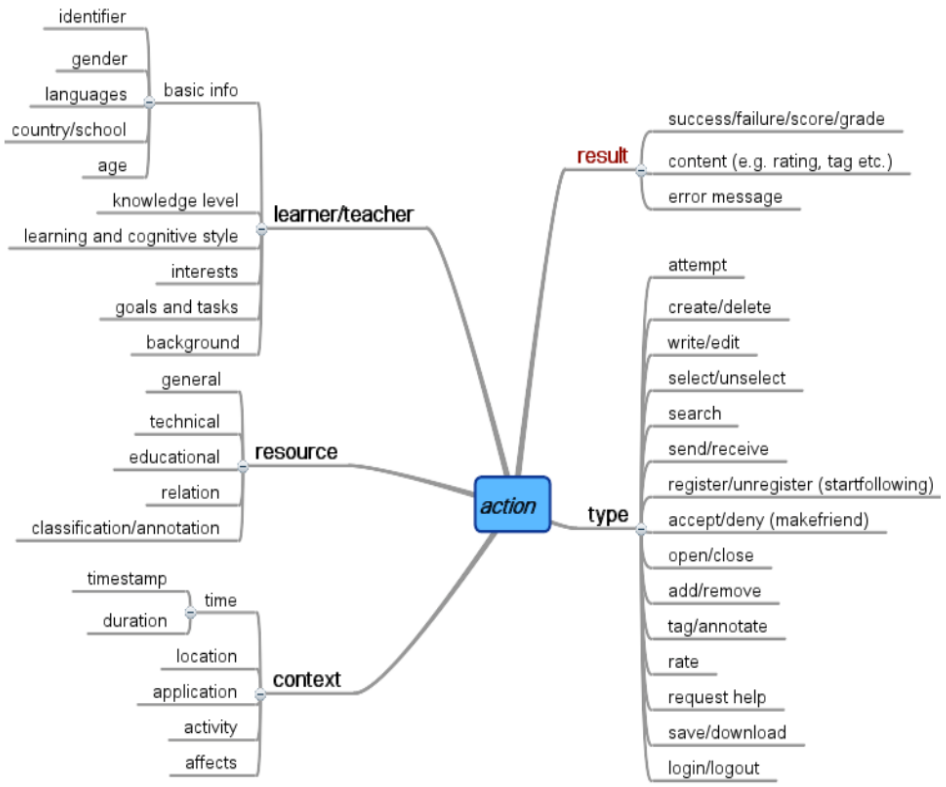


Figure 3.2.: Le Learner Action Model, un modèle des traces d'apprentissage, qui représente les informations régulièrement contenues dans les traces (VERBERT et al., 2012, p. 137).

que ces questions sont principalement d'ordre technique, cette dimension "*Comment*" est impactée par les trois autres dimensions.

Ainsi, la dimension "*Qui/Pour qui*" propose de s'intéresser aux personnes concernées par l'analyse et à leur compétences, ainsi qu'aux différentes réglementations et contraintes en vigueur (*e.g.* éthique, confidentialité). Dans le cadre des traces, *via* la dimension "*Quoi*", il convient de prendre en compte la source des traces et l'environnement associé. De plus, dans cette dimension, l'accent est également mis sur la question de la qualité des traces (*e.g.* données manquantes ou mal exportées). Enfin, à travers la dimension "*Pourquoi*", ce modèle invite à s'intéresser au besoin d'analyse et aux raisons d'un tel besoin. Cette dimension fait intervenir des questions concernant les résultats recherchés, comme les indicateurs.

Ces différentes dimensions font ressortir le besoin de tenir compte du contexte pédagogique lors des analyses de traces. Or, comme nous l'avons vu, les outils d'analyse actuels ne sont principalement orientés qu'autour de considérations techniques et computationnelles et ne permettent pas de tenir compte efficacement de ces spécificités pédagogiques – encore moins d'une manière unifiée (COOPER, 2013). Il en resulte que les informations liées aux différentes étapes de l'analyse des traces ne sont pas correctement représentées et qu'il est complexe d'identifier celles relèvant des contraintes techniques et celles relevant des contraintes pédagogiques.

Toutefois, UTL, *via* son modèle DGU, permet de représenter dans les processus d'analyse certaines des informations qui sont suggérées par le modèle de référence proposé par Chatti & *al.* (CHOQUET et IKSAL, 2007). Par exemple, la partie UTL/S, visible dans la Figure 2.10 (cf. Chapitre 2 p.22), permet d'expliquer les objectifs d'observations motivés par les objectifs pédagogiques. Le modèle DGU permet quant à lui d'apporter des informations contextuelles supplémentaires. Par exemple, la manière dont utiliser un indicateur peut être explicitée, tout comme à qui se destine un tel indicateur.

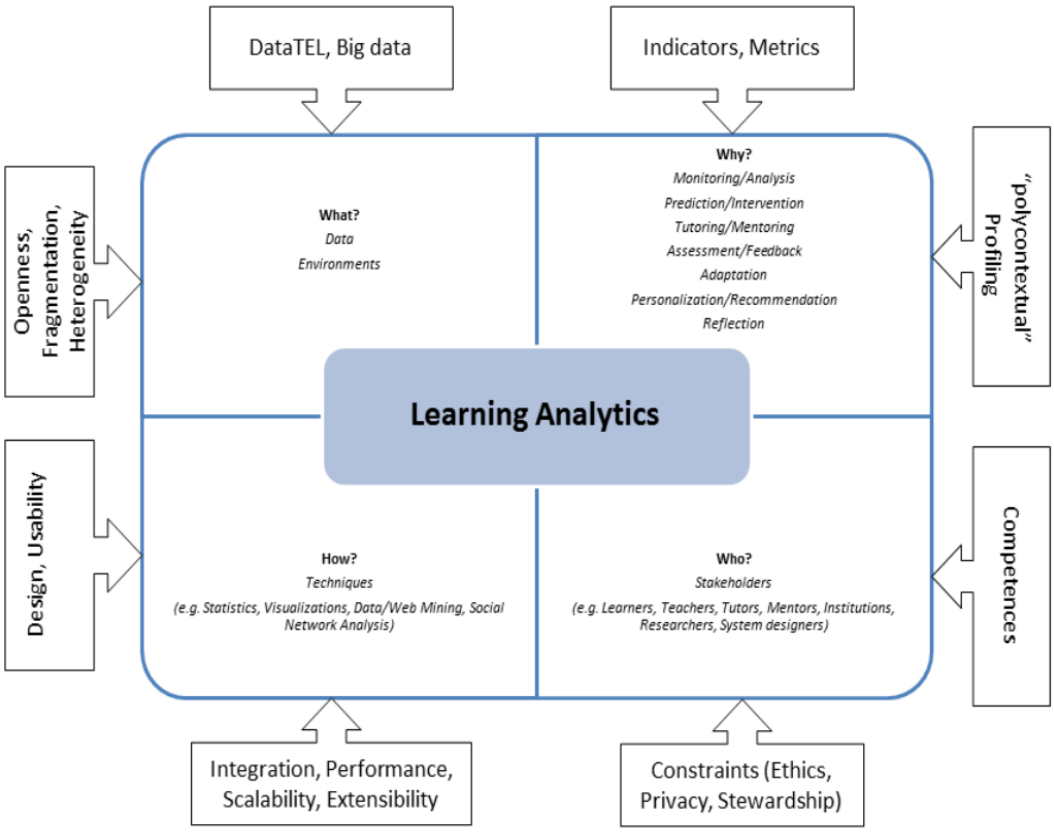


Figure 3.3.: Un modèle de référence pour les Learning Analytics prenant en compte des aspects techniques et contextuels (CHATTI et al., 2012, p. 7).

Pour conclure, ces travaux apportent une meilleure compréhension de l'impact du contexte sur la manière d'analyser les traces et apportent déjà quelques pistes pour s'en saisir. Ces travaux renforcent ainsi la perspective d'une capitalisation des processus d'analyse de traces d'apprentissage. Ils permettent de renforcer la réutilisation des processus d'analyse et de favoriser leur adaptation à d'autres situations pédagogiques, notamment par l'utilisation d'informations supplémentaires (J. C. STAMPER et al., 2011). Néanmoins, ces informations sont utilisées en annexes des processus d'analyse, comme on l'a vu. La conséquence est qu'elles ne permettent pas d'expliciter directement l'impact du contexte sur les processus, les choix de mise en œuvre effectués ou encore le paramétrage. En outre, il n'existe pas, à notre connaissance, de consensus fort dans nore domaine sur la manière de représenter le contexte et son effet.

Dans la section suivante, nous nous intéressons aux travaux en lien avec la capitalisation qui existent dans d'autres disciplines.

3.2 Capitaliser hors de la communauté EIAH

Constatant que les solutions actuelles de la communauté EIAH ne sont pas suffisantes pour permettre la capitalisation des processus d'analyse de traces, nous nous intéressons dans cette section aux efforts réalisés hors de cette dernière. Nous présentons des travaux majeurs issus de différents domaines à propos des processus d'analyse – qui ne concernent plus uniquement les traces d'apprentissage. Ces travaux ont permis de faire des avancées à propos de certaines propriétés utiles à la capitalisation des processus d'analyse, comme leur partage, leur intéropérabilité ou encore leur réutilisation.

Nous présentons tout d'abord les standards d'intéropérabilité et de portabilité proposés par le Data Mining Group, en particulier PMML. Puis, nous étudions les efforts de mise en commun et de

réutilisation dans la discipline des workflows, avant de nous intéresser brievement à la modélisation des processus métiers et de l'information qui peut y être représentée. Nous terminons cette section en explorant comment la reproduction et la qualité des analyses sont abordées dans le cadre des notebooks.

3.2.1 Les travaux du Data Mining Group

Predictive Model Markup Language

PMML (Predictive Model Markup Language) est développé par le Data Mining Group (DMG) (DATA MINING GROUP, 2018[a]), un consortium indépendant motivé par les besoins des vendeurs. Il s'agit d'un standard libre basé sur le langage XML, dédié à la représentation des modèles de data mining, et plus particulièrement des modèles statistiques et prédictifs (DATA MINING GROUP, 2018[j]). L'objectif affiché est de promouvoir le partage de tels modèles entre différents outils d'analyse en évitant l'incompatibilité des méthodes utilisées et en évitant les problématiques liées à l'utilisation de technologies propriétaires.

Proposé en 1999 (GROSSMAN et al., 1999), l'on constate aujourd'hui que PMML jouit d'un statut de langage commun pour le partage de solutions prédictives parmis les différents outils d'analyse. Ce standard est utilisé (partiellement) par plus d'une quarantaine d'outils d'analyse, comme Knime ou SPSS (DATA MINING GROUP, 2018[g]). Cela s'explique par la possibilité de ce standard à représenter, en plus des modèles utilisés, les données d'entrée nécessaires ainsi que certaines transformations qu'il est nécessaire d'appliquer sur ces données.

Un document PMML décrit un modèle de data mining obtenu dans un outil d'analyse quelconque. Pour ce faire, le document est constitué de différents éléments qui encapsulent des fonctionnalités spécifiques en fonction qu'il s'agisse du modèle en lui-même, des données d'entrée ou de sortie. La structure générale d'un document PMML est composée de son en-tête (*Header*), d'un *Data Dictionary*, de *Data Transformations* (issue d'un *Transformation Dictionary*), du modèle (*Model*), du *Mining Schema*, et des cibles (*Targets*) (GUAZZELLI et al., 2009b). La Figure 3.4 montre cette structure et illustre comment un modèle complexe peut être représenté avec PMML.

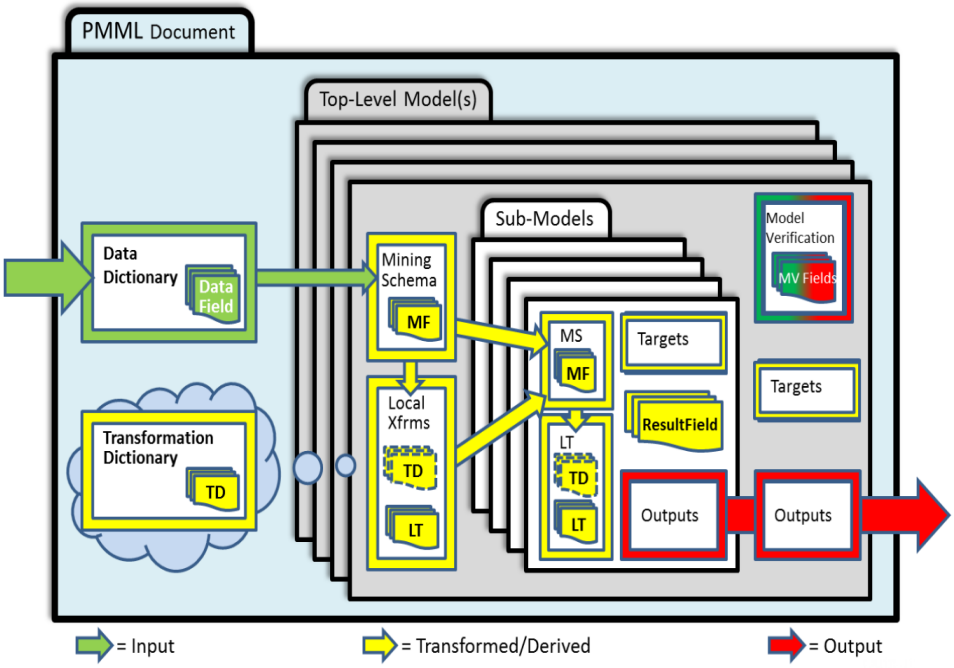


Figure 3.4.: Illustration d'un document PMML, montrant différents éléments pouvant intervenir dans la définition des modèles et sous-modèles, ainsi que leurs relations (DATA MINING GROUP, 2018[h]).

L'entête contient des informations générales sur le document PMML (*e.g.* son copyright), mais aussi des détails techniques en lien avec l'environnement d'où provient le modèle représenté par ce document. Le *Data Dictionary* contient les définitions et les propriétés de toutes les variables utilisées par le modèle. Ce dictionnaire permet d'indiquer par exemple le type des variables (*e.g.* entier), si elles ont des domaines de validité spécifiques ou encore si elles sont continues ou non. Ces définitions sont indépendantes des jeux de données utilisés pour entraîner le modèle et permettent également d'adopter un comportement par défaut en cas de valeur manquante dans les entrées (*Input*, la flèche verte à l'extrémité gauche de la Figure 3.4) (DATA MINING GROUP, 2018[j]).

Le *Transformation Dictionary* contient toutes les transformations qu'il est possible de représenter dans PMML. Une transformation, au sens de PMML, est une mise en correspondance de données vers une structure de données précise, adaptée pour être utilisée dans un modèle précis. Ces transformations sont représentées par les flêches jaunes centrales dans la Figure 3.4. On peut citer par exemple des transformations de discrétisation ou encore d'aggrégation des données. La possibilité de représenter ces transformations en conjonction avec les paramètres qui définissent les modèles eux-mêmes est considérée comme un concept clef de PMML (GUAZZELLI et al., 2009b). Cela permet en effet de s'assurer que l'entièreté du modèle est partagé.

Le *Model* est la partie centrale de PMML (les containers labélisés *Top-Level Model(s)* et *Sub-Models* de la Figure 3.4), puisqu'il contient toute la définition du modèle de data mining. Les attributs courants qui le constituent sont soit le nom du modèle, le nom de la fonction ou celui de l'algorithme utilisés. Ces attributs varient cependant en fonction du modèle dont il est question. Il est de plus important de noter que la représentation et la structure de chaque modèle (*e.g.* réseaux neuronaux, régression) est fixe et standardisée en amont par le *Data Mining Group*.

Ainsi, pour détailler le comportement spécifique à adopter pour un modèle, PMML s'appuie sur la fonctionnalité *Model Specifics*. Cela revient à représenter les configurations obtenues après avoir entraîné le modèle dans un outil d'analyse quelconque. Par exemple, un arbre de décision est représenté en PMML par ses noeuds, en suivant une structure récursive. Pour chaque noeud est associé un test, la variable testée et les noeuds fils qui en résultent. De plus, en préambule de tout modèle est défini un *Mining Schema*, qui définit toutes les variables du modèle. Il doit être respecté pour pouvoir utiliser le modèle en question. S'ensuit la représentation du modèle lui-même.

PMML offre donc un cadre technique solide pour pouvoir partager et réutiliser des modèles issus de techniques de data mining et de statistiques. En effet, en se positionnant comme langage pivot, PMML permet d'affranchir les modèles des contraintes techniques générés par les outils. Ainsi, bien qu'à notre connaissance il n'existe pas d'entrepôt de documents PMML comme il existe des entrepôts de processus d'analyse (cf. UnderTracks (MANDRAN et al., 2015) ou Tigris (J. STAMPER et al., 2016)), des travaux exploitent l'intéropérabilité des modèles PMML pour offrir des solutions de calcul décentralisés, jouant le rôle de plateformes génériques de calcul. C'est par exemple le cas d'ADAPA (GUAZZELLI et al., 2009a), une plateforme implémentant entièremment le standard PMML.

En outre, PMML représente un formidable travail de formalisation des modèles prédictifs et statistiques de data mining. Depuis près de 16 ans, les modèles décrits sont affinés à travers les différentes versions de PMML et de nouveaux modèles sont également proposés (*e.g.* Réseaux Bayesien dans la version 4.3) (DATA MINING GROUP, 2018[f]), pour correspondre aux besoins de la fouille de données. Ces évolutions sont possibles grâce à la position centrale de PMML entre organisations, utilisateurs finaux et chercheurs.

Cependant, il faut noter que PMML se destine à représenter les modèles de data mining et ne permet pas de représenter tous les opérateurs pouvant survenir lors d'un processus d'analyse (*e.g.* un filtre temporel). Ainsi, PMML ne se destine pas à représenter un processus d'analyse dans son ensemble, mais uniquement le modèle entraîné en résultant – ce qui exclut de surcroît la création d'indicateurs.

Il est aussi à noter que la définition d'un nouvel élément dans le standard PMML (*e.g.* modèle, transformation) est un long processus, administré par le groupe de travail de ce standard – imputable à sa position de standard. Par ailleurs, l'adoption de PMML est laissée à la discrétion des organisations développant les outils d'analyse. Comme nous l'avons vu, PMML n'offre pas une solution computation-

nelle directe mais définit uniquement comment représenter les modèles : c'est aux outils d'analyse d'être techniquement capables d'interpréter cette représentation et de l'exécuter dans leur écosystème. Il en résulte que la majorité des outils supportant PMML ne le font que partiellement et ne supportent que certains modèles, ou bien ne suivent pas l'évolution du standard (DATA MINING GROUP, 2018[g]). La conséquence directe est que, dans ces outils, la réutilisation et le partage des modèles est ainsi réduit.

Enfin, PMML ne permet pas de faciliter l'adaptation de l'analyse à d'autres besoins d'analyse. En effet, il ne permet pas de représenter les informations techniques qui ont mené à l'élaboration du ou des modèles représentés (*e.g.* le choix des données d'entraînement), ni les informations contextuelles de l'analyse. En sus, puisqu'un document PMML représente un modèle entraîné, il n'est pas possible d'y appliquer des modifications directement (DATA MINING GROUP, 2018[b]). Or, si, lors de la réutilisation d'un document PMML, les résultats ne sont pas satisfaisants, il ne sera pas possible d'appliquer des modifications directement au document et de s'assurer que ces changements sont pertinents.

Portable Format for Analytics

PFA, pour Portable Format for Analytics, est qualifié par le DMG comme un standard émergent pour les services exécutant des modèles statistiques et des transformations de données (DATA MINING GROUP, 2018[i]). Il s'agit d'un langage prévu pour assurer la transition des analyses d'un environnement de dévelopemment dans un environnement de production dédié, en facilitant leur partage, leur mise à l'échelle et en assurant, *a contrario* de PMML, une certaine flexibilité algorithmique (DATA MINING GROUP, 2018[b]). L'objectif est de renforcer l'analyse en tant que telle, en y limitant les dépendances et en proposant une structure facilement maintenable et plus résistante aux erreurs (DATA MINING GROUP, 2018[b]).

PFA se destine à être un "mini-langage" de calcul mathématique interprétable pour transformer les données (DATA MINING GROUP, 2018[c]). Ces transformations sont décrites au sein de documents PFA, écrits en JSON. Ces documents PFA peuvent être produits par d'autres outils d'analyse et définissent des *scoring engines*. Un *scoring engine* est composé d'un triplet $\langle Input - Action - Output \rangle$, où un ensemble de fonctions (*i.e. Action*) sont utilisées sur les données en entrée (*i.e. input*) pour construire les données de sortie (*i.e. output*) (DATA MINING GROUP, 2018[c]). PFA est ainsi plus proche d'un langage de programmation "classique" que peut l'être PMML, en ceci qu'il définit les instructions à réaliser avec un niveau de granularité plus important.

Puisqu'il est possible de chaîner et de combiner ces fonctions et les *scoring engines*, PFA peut s'appliquer aux différentes étapes de l'analyse, et non pas uniquement aux modèles de data mining. Il en résulte que les étapes de prétraitements et de post-traitements de l'analyse peuvent ainsi être définies dans PFA (PIVARSKI et al., 2016). Avec ces propriétés, PFA adopte une approche par pipeline de données[2] : il ne se destine pas à opérer directement avec un environnement technique, ni à en dépendre.

Aussi, PFA se destine a être déployé et exécuté comme une machine virtuelle au sein d'envrionnements de traitement appelés hôtes PFA (DATA MINING GROUP, 2018[d]). L'objectif est d'observer une indépendance technique forte entre l'analyse (*i.e.* des triplets $\langle Input - Action - Output \rangle$ issus d'outils d'analyse comme R), et le système qui l'exécute et gère le pipeline (*e.g.* une architecture Hadoop (WHITE, 2009)). Du point de vue d'un tel système, un document PFA peut être vu comme un fichier de configuration, ce qui permet d'envisager une évolution de l'analyse indépendamment du pipeline en question (DATA MINING GROUP, 2018[b]).

PFA a plusieurs avantages par rapport à PMML, malgré une complexité technique plus importante. Une des contraintes fortes qu'impose PFA est que deux systèmes qui exécutent du PFA doivent produire avec le même jeu de données en entrée les mêmes résultats (DATA MINING GROUP, 2018[c]). Il permet par exemple de représenter des structures de contrôles, comme des conditions ou des fonctions décrites par l'utilisateur, améliorant les possibilités de description de l'analyse. De plus, la granularité des fonctions disponibles est suffisament fine pour permettre à l'utilisateur de définir plusieurs chaînes de fonctions, ainsi que de gérer leur séquence d'exécution. Il est également possible pour les modèles

2. data pipeline

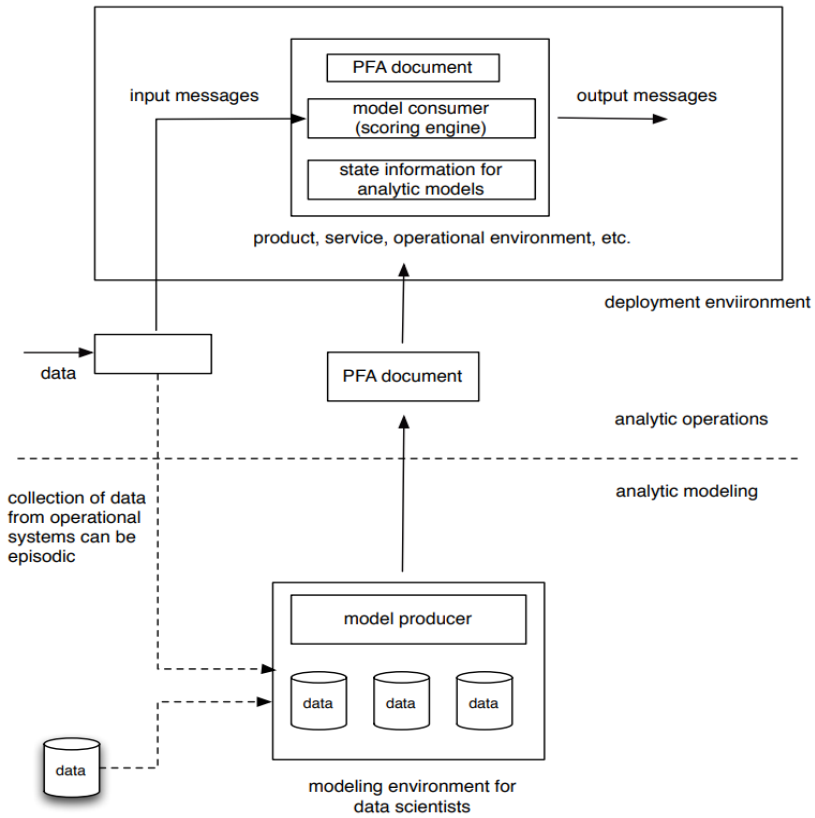


Figure 3.5.: Schéma illustrant l'échange et la mise en œuvre par le *model consumer* dans un hôte PFA de modèles préalablement élaborés par le *model producer* (PIVARSKI et al., 2016, p. 6).

de partager et d'utiliser des données externes, par exemple provenant d'une base de données tierce (PIVARSKI et al., 2016), comme on peut le voir dans la partie inférieure de la Figure 3.5.

Toutes ces propriétés font de PFA un candidat intéressant pour le partage et la mise à l'échelle des analyses. De plus, le cadre que fournit ce standard permet de faire évoluer les analyses indépendamment de l'environnement technique. Néanmoins, PFA décrit – à la différence de PMML – le comportement "bit à bit" exact des transformations à appliquer sur les données, indépendamment de l'environnement (DATA MINING GROUP, 2018[c]). Cela conditionne le type d'environnements dans lequel PFA peut être déployé et limite de surcroît l'intéropérabilité des analyses. De plus, cette précision technique s'obtient en contrepartie des possibilités de description de l'information en lien avec l'analyse (*e.g.* les choix d'implémentations, les dépendances contextuelles), et limitent la compréhension des analyses et leur adaptation. D'après le DMG, PFA est difficile à lire ; il se destine plus à être manipulé programmatiquement qu'à la main (DATA MINING GROUP, 2018[e]).

3.2.2 Les travaux autour de l'e-Science

Le domaine de l'e-Science a trait à l'analyse de données massives, principalement dans des environnements fortement distribués. Ce domaine s'intéresse de plus à toutes les étapes de la démarche de recherche, de la définition de la problématique de recherche à la publication des résultats et de la méthodologie utilisée (IEEE, 2018). Dans ce domaine, nous observons que les workflows se sont progressivement imposés comme un outil populaire, intuitif et efficace, pour répondre à ces attentes concernant l'analyse des données (GIL et al., 2007). Cette adoption massive s'explique entre autres par la capacité des workflows scientifiques à représenter la démarche scientifique – principalement l'analyse des données, ainsi qu'à la manière de traîter ces données (*e.g.* invocation de services distants) (DEELMAN et al., 2009).

Bien que les workflows scientifiques sont des entités complexes et coûteuses à mettre en œuvre, ils représentent un réel intérêt sur le plan scientifique. Ils font en effet état d'une expertise scientifique souvent tacite dans l'analyse des données et peuvent de plus être considérés comme des protocoles expérimentaux. Ils explicitent les étapes précises qui doivent être suivies dans le cadre de l'analyse, la rendant répétable voire, dans le meilleur des cas, reproductible (C. A. Goble et D. C. De Roure, 2007). Ces workflows ont rapidement été partagés pour permettre leur réutilisation par d'autres scientifiques (Wroe et al., 2007). Pour cela, un effort conséquent a été réalisé pour les considérer comme *ethos*, autrement dit des entités scientifiques à part entière, au même titre que les données et les articles de recherche (C. A. Goble et D. C. De Roure, 2007).

Ce travail de reconnaissance s'observe particulièrement dans le cadre de la biologie, où des propositions importantes ont été faites pour soutenir le développement de tels workflows, comme le projet ^{my}Grid (myGrid, 2008 ; Addis et al., 2003) (désormais eScience Lab (eScience Lab Research Group, 2018)). Des outils dédiés à la gestion de ces workflows, appelés *scientific workflow management system* (SWfMS), ont émergé afin d'assister les scientifiques par l'automatisation de la récolte et de la gestion des données, de leur provenance, ainsi que de leur traitement (Altintas et al., 2004 ; Oinn et al., 2006). Ces SWfMS se différencient d'outils de data mining classique, comme Orange : Data Mining ou Knime, en cela qu'ils considèrent le cycle de vie de l'e-Science et tentent d'y placer en son centre le scientifique et son contexte expérimental (Oinn et al., 2006 ; Bánáti et al., 2016).

Pour constater cela, on peut citer par exemple Taverna, un SWfMS proposé dans le cadre du projet ^{my}Grid (Oinn et al., 2004). En plus de gérer et d'exécuter les workflows, cet outil permet d'invoquer divers services de traitement des données (*e.g.* des services SOAP ou REST Web), et ainsi favoriser l'interopérabilité avec d'autres workflows développés dans des outils différents. Taverna permet également de tenir compte des informations de provenance des données importées et produites, ce qui permet d'exposer des détails du workflow et d'en examiner son exécution. En plus de ces métadonnées de provenance, Taverna propose de stocker des métadonnées supplémentaires concernant le workflow et les données présentes, notamment des métadonnées d'identification (*e.g. Life Science Identifer (LSID)*). Taverna se dote grâce à ces métadonnées de la possibilité d'une recherche sémantique fondée sur ces identifiants, à travers l'invocation d'un service externe. Enfin, cet outil rend possible l'intégration d'informations en lien avec l'expérimentation et la recherche menées, comme nous le verrons dans la Section 4.1 avec les *Research Object*.

Tous ces travaux qui ont apporté une formalisation plus importante des workflows scientifiques (Bánáti et al., 2015), et qui les ont considérés comme des ressources à part entière, ont permis l'émergence d'environnements de recherche virtuels (C. A. Goble et D. C. De Roure, 2007 ; D. De Roure et al., 2009 ; Goecks et al., 2010 ; Mates et al., 2011). Ces environnements permettent, à l'instar de travaux comme Tigris (J. Stamper et al., 2016) ou UnderTracks (Mandran et al., 2015) de notre communauté, de stocker et de partager les workflows créés. Néanmoins, ces outils en diffèrent en considérant les workflows comme des ressources scientifiques particulières et en amplifiant l'aspect collaboratif entre les différents acteurs (*e.g.* chercheur, utilisateur final), ceci dans l'objectif de renforcer la qualité et la reproductibilité des analyses menées (D. De Roure et al., 2009).

La particularité de ces environnements tient du fait qu'ils s'inscrivent dans une démarche majoritairement sociale (C. A. Goble et al., 2010). L'interaction entre les différents acteurs est accrue et la quantité d'informations disponibles en lien avec les workflows et la problématique de recherche associée y est plus importante. Il en résulte une nécessité accrue de comprendre les ressources qui sont mises à disposition de la communauté, afin qu'elles puissent être réutilisées convenablement : par conséquent, cela nous donne des pistes pour la capitalisation des processus d'analyse.

Dans ^{my}Experiment (myExperiment, 2015a) par exemple, chaque workflow possède une zone de discussion dédiée, diverses métadonnées (*e.g.* des tags, licence) et ressources (*e.g.* les données initiales, des schémas), ainsi que des informations associées à ses aspects techniques (*e.g.* opérateurs utilisés, services externes dont il dépend). Il est également possible de poster une évaluation du workflow, et d'en proposer une nouvelle version. De plus, certains de ces outils, comme Galaxy (Goecks et al., 2010), permettent directement la création, la modification et l'exécution de workflows dans l'environnement, offrant ainsi un écosystème dédié à l'e-Science.

Pour conclure, le partage et la réutilisation des workflows scientifiques semble être soutenus à la fois par l'abstraction que permet l'approche par workflows (Oinn et al., 2006), et par l'aspect social véhiculé par ces outils, comme nous permet de le constater leur forte adoption au sein de leur communauté (Galaxy Project, 2018 ; MyExperiment, 2015b). Ces environnements de recherche favorisent ainsi l'accessibilité des analyses, leur reproductibilité mais aussi leur ouverture, puisqu'il devient possible par exemple de directement référencer le protocole dans des publications scientifiques (Goecks et al., 2010). Ils constituent ainsi des pistes intéressantes pour la capitalisation des processus d'analyse de traces d'apprentissage.

Néanmoins, ces approches – fortement techniques – font apparaître le problème important de la dégradation des workflows dans le temps, et s'y confrontent (D. De Roure et al., 2011). Ces workflows reposent sur des ressources *ad-hoc* ou externes pour mener à bien ces analyses qui, avec le temps, peuvent être modifiées, altérées, supprimées ou simplement inaccessibles, empêchant ainsi leur bonne exécution. Cette dégradation s'explique majoritairement par le fait que les workflows manquent d'informations, à la fois techniques et contextuelles (*e.g.* meta-données sur la provenance des paramètres), pour permettre leur adaptation. Plus précisément, Zhao & *al.* (Belhajjame et al., 2012b) identifient l'évolution des ressources externes, le manque d'exemples de données, l'insuffisance des environnements d'exécution et le manque de métadonnées comme les quatre causes principales de cette dégradation. En ce sens, des alternatives existent pour palier ce problème, comme nous le verrons en Section 4.1.

3.2.3 Vers d'autres initiatives participant à la consolidation de la capitalisation

Toujours dans une approche orientée workflow, un autre travail intéressant relatif à la capitalisation est la norme *Business Process Model & Notation* (BPMN) (Object Managment Group, 2011), maintenue par l'Object Management Group. Cette norme sert à décrire les activités métier pouvant survenir au sein d'une organisation. Bien que partageant superficiellement des points communs avec les workflows scientifiques, ils sont conçus principalement pour le calcul de données plutôt que pour le contrôle des activités (Altintas et al., 2004). Les workflows d'activités modélisés à partir de BPMN permettent d'apporter une abstraction supplémentaire sur l'activité décrite et de la décomposer de manière formelle.

BPMN, ainsi désintéressée de l'aspect computationnel des données, permet de mettre en avant, par rapport aux workflows scientifiques, des éléments nouveaux qui trouvent un écho au sein de notre communauté et de l'analyse de traces. Ainsi, nous pouvons constater que BPMN permet de modéliser l'orchestration des tâches avec une granularité fine, enrichie par un système de conditions et d'événements communiquant entre les différentes tâches du workflow. Il fait également apparaître la notion d'acteur dans le workflow, et permet ainsi d'associer des rôles spécifiques à des tâches précises. En outre, il permet de représenter la collaboration qui peut exister entre différentes entités impliquées dans l'activité, ce qui fait écho à l'aspect pluridisciplinaire de notre communauté.

Enfin, pour conclucre ce chapitre, nous pouvons noter les travaux en lien avec les *notebooks* informatiques. Ce sont des programmes constitués à la fois de texte libre, généralement sous un format de balisage léger (*e.g.* markdown), et de code source écrit dans un langage (*e.g.* Python, C++), destinés à l'analyse de données. L'un de ces principaux représentants est Jupyter (Kluyver et al., 2016), qui est une évolution du projet IPython (Perez et Granger, 2007). L'objectif de ces *notebooks* est également de favoriser l'exploration interactive des données, le partage et la reproductibilité des analyses menées.

Ce faisant, les notebooks adoptent une démarche que nous qualifions de *narrative*. Ils sont assimilables à un document où le code de l'analyse est découpé en blocs logiques, qui peuvent être individuellement modifiés et exécutés, et où il est possible de fournir des informations entre chacun de ces blocs, sous forme de texte libre. Il est également possible d'introduire d'autres éléments dans ces documents, comme des formules mathématiques ou des figures qui peuvent évoluer dynamiquement en fonction de la configuration des blocs de codes.

Grâce à cela, il est possible de tenir à la fois compte des informations techniques et contextuelles en lien avec l'analyse et de représenter le savoir faire mis en œuvre. Néanmoins, bien que cette approche constitue un support supplémentaire à la réutilisation et à l'adaptation par sa souplesse, elle ne s'effectue qu'à partir d'éléments non structurés, comme du texte libre : il n'existe pas, à notre connaissance, de modélisation. Les notebooks ne possédant peu, voire pas, de métadonnées, les informations n'y sont donc pas structurées et elles peuvent être complexes à interpréter et à manipuler par la machine.

4 Comment structurer et utiliser l'information relative aux processus d'analyse ?

Sommaire

Section	**Introduction**	**41**
Section 4.1	**Modélisation sémantique**	**42**
4.1.1	De la modélisation du processus d'analyse...	42
4.1.2	... À la modélisation de nouvelles informations	50
Section 4.2	**Interpréter et exploiter la sémantique à travers les requêtes utilisateurs**	**55**
4.2.1	Une étude des capacités des outils d'analyse actuels	56
4.2.2	Approximation des requêtes	59

Introduction

Les processus d'analyse, qu'ils soient portés sur des traces d'apprentissage ou non, sont des artefacts complexes par nature, qui font état d'une expertise humaine déployée pour répondre à un besoin. Par leur implémentation dans des outils d'analyse, cette expertise s'en retrouve figée du fait des spécificités inhérentes à ces outils. Néanmoins, comme nous avons pu le constater précédemment, il n'existe pas de consensus entre ces outils d'analyse ; la représentation de l'analyse dans son ensemble, comme ses opérations, ses configurations ou encore ses hypothèses émises, n'est pas uniformisée. Dans certains cas, il s'avère même que certains éléments ne peuvent pas être représentés. Cette diversité est à l'origine d'une ambiguïté latente, qui peut se manifester notamment lorsque l'on cherche à comprendre ou à réutiliser un processus partagé.

En outre, ce manque de formalisme pose des difficultés pour rendre les processus d'analyse interprétables par la machine, que ce soit sur le plan computationnel (*e.g.* deux opérateurs différents portant le même nom) que sur le plan du raisonnement, c'est-à-dire raisonner avec les informations formant la mise en œuvre de l'analyse (*e.g.* métadonnées, ordre des opérations) pour inférer des connaissances. Cela contribue à proposer des solutions *ad-hoc* d'exécutions, d'assistances (*e.g.* gestion des erreurs d'appel de services externes dans Taverna (Oinn et al., 2006)) et de recherches des analyses, qui ne sont, par définition, pas généralisables à d'autres outils. Ce constat réalisé au sein de différentes communautés a motivé la proposition de modélisations sémantiques, dans l'objectif de remédier à l'ambiguïté de l'information et permettre son interprétation par la machine.

Dans ce chapitre, nous nous intéressons à ces modélisations sémantiques, souvent sous forme d'ontologies, et aux possibilités qu'elles offrent. Tout d'abord, nous présentons différentes ontologies qui permettent de définir les processus d'analyse, principalement leur aspect technique. Ensuite, nous nous intéressons à la manière dont ces ontologies peuvent représenter l'information non technique des processus d'analyse. Enfin, nous explorerons comment les outils actuels qui exploitent des ontologies en lien avec les processus d'analyse se servent de cette sémantique pour répondre aux requêtes utilisateurs et leur proposer divers services. Cette dernière partie nous permet d'ailleurs d'aborder des travaux issus de la discipline du web sémantique sur la manière de traiter et d'approximer des requêtes utilisateurs dans l'objectif de comprendre ces requêtes, puis d'assister ces utilisateurs.

4.1 Modélisation sémantique

Comme nous l'avons constaté dans le Chapitre 3, différents travaux de modélisation des processus d'analyse et des opérations existent et contribuent à renforcer certains aspects de la capitalisation, à l'instar de PMML (Data Mining Group, 2018[j]). Toutefois, ces modélisations n'abordent pas une démarche holistique de la mise en œuvre de l'analyse, et ne permettent pas de tenir compte de son aspect pluridisciplinaire. Elles se concentrent sur les propriétés computationnelles de l'analyse, en y déployant leurs propres terminologies, représentations et relations des contenus, et en y instaurant des règles d'exécutions spécifiques (*e.g.* PFA (Data Mining Group, 2018[i])). Or, ce manque d'information et leur caractère disparate causent des limitations quant à la reproduction et la réutilisation des analyses (Belhajjame et al., 2012b). En sus de cela, il faut aussi rappeler que le partage et la publication de ces analyses ont des besoins importants concernant certaines informations comme la qualité, la provenance ou les méthodes, tout comme c'est le cas pour les données que ces analyses exploitent (Bechhofer et al., 2013 ; Bienkowski et al., 2014 ; Mandran et al., 2015).

Une solution pour modéliser ces éléments complexes réside dans les ontologies. D'un point de vue informatique, et d'une manière générale, une ontologie est un modèle décrivant une certaine réalité, qui relie des concepts, exprimés par des termes. Les concepts reliés sont, en informatique, appelés des classes, ceux qui les relient sont appelés des propriétés. Une ontologie possède un vocabulaire clairement défini contenant tous ces termes, ainsi qu'un ensemble d'hypothèses explicites concernant la signification de ces termes (Guarino, 1998). Par exemple, dans les cas les plus simples, une ontologie peut se manifester comme un simple ensemble hiérarchique de concepts.

L'avantage de ces ontologies est qu'il devient possible de modéliser ladite réalité (*e.g.* les processus d'analyse) sous forme de propositions complexes, généralement avec de la logique du premier ordre, et de les rendre interprétables par la machine : il est ainsi possible de valider formellement les modélisations (Staab et Studer, 2010 ; Praxademia, 2015). Ces propositions complexes sont exprimées *via* des langages standardisés comme RDFS (W3C, 2014a) et OWL (W3C, 2012). Les ontologies favorisent ainsi la réutilisation des éléments qu'elles décrivent et leur interopérabilité (Gruber, 1993), tout en limitant l'ambiguïté sémantique.

Dans cette section, nous présentons différents travaux proposant des ontologies en lien avec les processus d'analyse. Nous nous intéressons d'abord à la manière dont l'information technique de ces processus d'analyse est structurée, comme la manière de représenter les étapes d'un processus d'analyse ou encore les données utilisées. Nous nous intéressons ensuite à la structuration de l'information contextuelle (*i.e.* non technique), ce qui nous permet en outre d'observer les éléments jugés pertinents de conserver dans un processus et la manière de les y intégrer.

4.1.1 De la modélisation du processus d'analyse…

Avant de considérer les travaux concernant la modélisation sémantique des processus d'analyse, remarquons que des initiatives visant à structurer sémantiquement les traces d'apprentissage sont apparues. Comme nous l'avons vu, différents standards et propositions existent et proposent une modélisation des traces, comme xAPI (Advanced Distributed Learning, 2013) ou Caliper (IMS Global Learning Consortium, 2018[e]). Toutefois, ces standards et propositions suggèrent certes un modèle de la trace, mais omettent souvent la sémantique associée à ce modèle. Les modélisations qui proposent un vocabulaire contrôlé et des métadonnées supplémentaires, comme c'est le cas avec xAPI et Caliper, sont *in fine* confrontées au problème que les métadonnées ainsi que le contenu des traces, voire aussi le vocabulaire, sont décrits en langue naturelle. La trace d'apprentissage ainsi modélisée peut potentiellement s'avérer ambiguë et difficilement interprétable par la machine (Advanced Distributed Learning, 2018[g]).

C'est pour tenter de répondre aux difficultés susmentionnées que ces initiatives adoptent une modélisation ontologique de la trace d'apprentissage. On voit ainsi apparaître, par exemple, une ontologie pour Caliper (IMS Global Learning Consortium, 2018[a]) et une pour xAPI (Advanced Distributed Learning, 2018[d]), avec un vocabulaire contrôlé et sémantiquement défini (Advanced Distributed

LEARNING, 2018[f]). Ces propositions reposent sur des techniques du web sémantique (*e.g.* RDF, OWL, URI), ce qui permet aux termes utilisés d'être déréférencés [1] manuellement ou de manière automatique par la machine, afin d'en connaître leur sens et leurs propriétés. Par exemple, dans xAPI, les termes (*i.e.* les classes) *xapi :Abandoned* et *xapi :Watched* sont subsumés par la classe *xapi :Verb*, signifiant qu'ils sont des verbes et qu'ils sont valides lorsqu'utilisés dans des triplets xAPI de la forme $\langle Acteur - Verbe - Objet \rangle$.

Néanmoins, ces ontologies ne se contentent pas de proposer une vision propre de la modélisation des traces. Un important effort a aussi été réalisé pour les aligner à d'autres ontologies déjà existantes et standardisées (VIDAL et al., 2015 ; DE NIES et al., 2015). Autrement dit, certains termes utilisés pour modéliser les traces sont désignés équivalents à d'autres contenus dans des vocabulaires différents. Par exemple, Vidal & *al.* (VIDAL et al., 2015) suggèrent que la classe *xapi :Actor* peut être alignée avec la classe *foaf :Agent* de l'ontologie Friend Of A Friend, qui permet de décrire des personnes ainsi que les relations qu'elles entretiennent entre elles (DAN et LIBBY, 2014). Il devient alors possible d'intégrer à xAPI des relations entre les différents acteurs de la trace.

Un autre exemple d'alignement est visible dans la Figure 4.1. Chaque élément d'un triplet xAPI est ici mis en correspondance avec un terme issu de l'ontologie PROV (LEBO et al., 2013) destinée à exprimer la provenance d'une entité – nous la décrivons page 44 de ce chapitre. Cet alignement permet ainsi de renforcer la sémantique des traces, et d'indiquer quel objet, au sens xAPI, a été utilisé lors d'une activité particulière réalisée par un agent (*i.e.* un acteur). L'une des conséquences directes de cet alignement est de renforcer la possibilité de publication et de partage des traces, puisqu'il devient possible de les stocker aussi dans des entrepôts de provenances plus généralistes. Une autre conséquence est aussi de pouvoir utiliser la sémantique issue de PROV pour interroger ces traces.

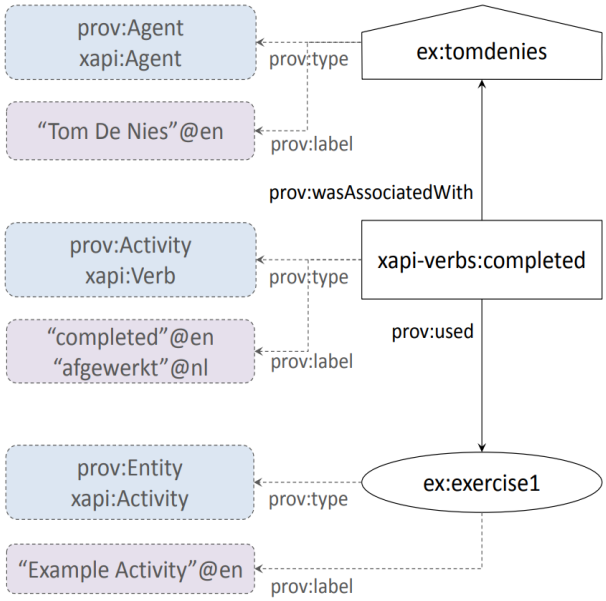


Figure 4.1.: Exemple d'un triplet xAPI (à droite) converti dans PROV (à gauche), d'après (DE NIES et al., 2015, p. 692).

Ainsi, ces ontologies contribuent à l'interopérabilité des traces, tout en renforçant la formalisation et les possibilités de raisonnements associés aux traces d'apprentissage (VIDAL et al., 2015). De plus, de telles ontologies réduisent potentiellement les *duplicata* du vocabulaire et favorisent sa réutilisation (ADVANCED DISTRIBUTED LEARNING, 2018[g]) ainsi que son extension. En effet, les nouveaux termes d'un vocabulaire peuvent être mis en relation avec d'autres, déjà présents, afin de définir et de renforcer la sémantique de ces anciens termes, et même la sémantique globale. Par exemple, un nouveau terme *custom :myAction* peut être subsumé par la classe *xapi :Verb* de xAPI pour indiquer qu'il s'agit d'un verbe et alors renforcer la validation des traces. Dans un contexte aussi complexe que celui

1. Requêter la représentation de la ressource sur laquelle pointe le terme.

de l'apprentissage, bien que l'élaboration de telles ontologies soit difficile, il n'en reste pas moins que nous pensons que ces possibilités constituent un atout important pour envisager la modélisation des traces d'apprentissage.

Remarquons toutefois qu'une majorité des travaux proposant des ontologies pouvant être mises en lien avec les processus d'analyse sont proches de ces initiatives de modèles sémantiques de traces d'apprentissage. Ces travaux de modélisation des processus adoptent une démarche centrée sur les données – ou plus généralement centrées entités – et cherche à en exprimer leur provenance. Ainsi, exprimer la provenance d'un résultat d'une analyse revient à être en mesure de représenter l'ensemble des opérations appliquées sur les données nécessaires à son obtention. De plus, si l'on cherche à exprimer la provenance des résultats intermédiaires de l'analyse, alors il est nécessaire de pouvoir modéliser chaque action de l'analyse qui les produit (Scheidegger et al., 2008).

Un premier travail allant dans ce sens est l'ontologie PROV (PROV-O) (Lebo et al., 2013). Ce travail du W3C (W3C, 2018) s'est établi comme un standard pour modéliser la provenance de diverses entités. Il permet d'exprimer des informations de provenances génériques à propos d'entités, d'activités et de personnes impliquées dans la production d'un artefact, dans l'objectif de pouvoir en évaluer sa qualité ou encore son authenticité (Moreau et al., 2013). La Figure 4.2 illustre comment l'ontologie PROV encode ce modèle de provenance.

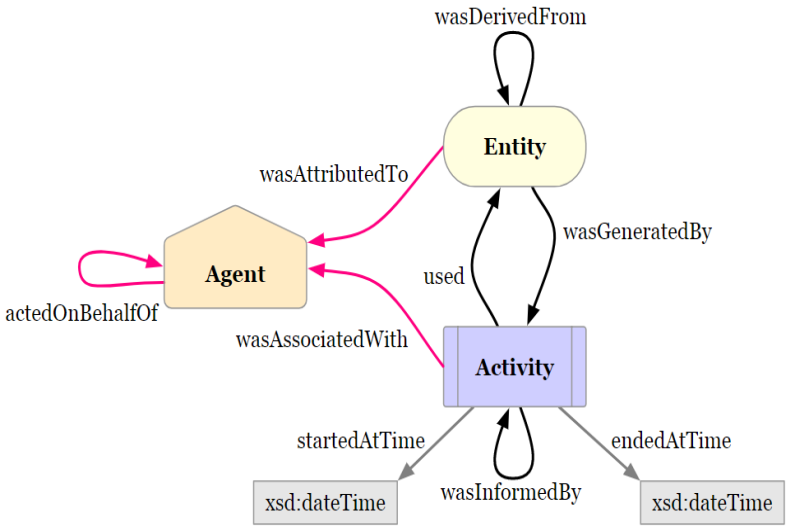


Figure 4.2.: Schéma de l'ontologie PROV-O, de ses trois classes principales et des relations qui existent entre elles (Lebo et al., 2013).

Ce modèle sémantique exprime la manière d'obtenir une entité (*Entity*) à partir d'une activité (*Activity*) réalisée par un agent (*e.g.* une personne, un système, une opération), et les propriétés qui existent entre ces éléments. Pour ce faire, une entité est définie comme étant produite par une activité (*via* la propriété *wasGeneratedBy*), et étant issue d'autre(s) entité(s) pré-existante(s) – cette relation étant modélisée par la propriété *wasDerivedFrom*. Pour chaque transformation, il est donc possible de modéliser les entités utilisées (propriété *used*) ; le chaînage de ces transformations peut être représenté par la propriété *wasInformedBy* (Lebo et al., 2013).

Du fait de sa généricité, PROV-O permet d'opérer une correspondance des processus d'analyse avec cette ontologie. L'analogie entre opérateur et activité s'opère – ou à plus gros grain entre le processus directement, tout comme l'analogie entre traces d'apprentissage et entités, où il est possible d'y distinguer les entrées et les sorties des opérateurs. La notion d'*agent* de PROV pouvant alors correspondre aux outils d'analyse utilisés, aux services distants nécessaires ou encore aux personnes menant l'analyse. Par conséquent, il devient possible d'exprimer des analyses mises en œuvre dans des contextes techniques différents et hétérogènes et de favoriser leur interopérabilité (De Nies et al., 2015).

PROV introduit en plus la notion de *plan* (*prov :plan*), qui est une entité ayant vocation à représenter un ensemble d'actions escomptées par un agent pour atteindre certains objectifs : il décrit comment une activité est effectuée. Un plan PROV n'est pas normatif et ne spécifie pas le contenu, ni les propriétés qu'il possède. Néanmoins, partager un plan peut présenter des bénéfices notables : faciliter la réutilisation d'un ensemble d'activités, décrire les attentes liées à l'exécution de cet ensemble d'activités, ou encore fournir une description plus abstraite de cette exécution (GARIJO et GIL, 2012).

Néanmoins, PROV-O n'offre qu'une démarche générique pour modéliser sémantiquement les processus d'analyse de traces décrits de cette manière. Cette ontologie ne permet pas de les modéliser entièrement (*e.g.* paramétrage des opérateurs), ni de tenir compte de leur complexité (BELHAJJAME et al., 2015). De ce fait, les processus modélisés de cette manière peuvent être enclins à des ambiguïtés (*e.g.* la notion d'agent pour une analyse, l'ordre opératoire des transformations sur les traces), en plus de ne pas pouvoir être interrogé avec les éléments sémantiques qui sont en lien avec l'analyse (FREIRE et al., 2008).

Partant de ce constat, des travaux ont proposé d'enrichir PROV-O avec des propositions spécifiques aux workflows. C'est par exemple le cas avec D-PROV (MISSIER et al., 2013), où les auteurs remarquent que le manque d'informations concernant la structure des workflows utilisés pour produire des résultats limite les perspectives de réutilisation. Alors que PROV peut, comme ils l'indiquent, répondre à des questions génériques du type "Quelles sont les activités conduisant à tel résultat ?", il n'est pas possible d'en connaire les paramètres, ni de s'assurer de la bonne structure du processus décrit ou encore de la manière dont il a évolué au cours du temps.

Ces informations de provenances non captées par PROV-O relatent des moments précis de l'activité. Et pour cela, elles sont qualifiées de rétrospective, de prospective et de "*self*-provenance" (FREIRE et al., 2008 ; LIM et al., 2010 ; MISSIER et al., 2013). La provenance rétrospective concerne la provenance des données produites lors d'une exécution d'un processus particulier ; la provenance prospective concerne la représentation même du workflow (*e.g.* son paramétrage) ; la "*self*-provenance" permet de tenir compte de l'évolution du processus en fonction des différentes versions. D-PROV permet de capturer ces informations et se destine ainsi à représenter les workflows exécutables.

Pour ce faire, D-PROV étend PROV avec des classes supplémentaires. Non exhaustivement, ces classes permettent de représenter la notion de workflow en lui-même, les tâches (*e.g.* opérations) survenant à l'intérieur de ces workflows, ainsi que les données d'entrée et de sortie. En plus de cela, elles permettent de représenter la connexion qui peut exister entre deux tâches, ce qui permet d'indiquer les données qui y sont échangées. D-PROV repose également sur un modèle de provenance hiérarchiquement structuré des workflows : l'avantage est ainsi de pouvoir imbriquer des worflows à l'intérieur d'autres lors de leur modélisation.

En cela, D-PROV permet de modéliser plus précisément la provenance des données issues de l'exécution des workflows, et certains de leurs aspects techniques, comme le type de communication des activités (*e.g.* port-à-port ou canal-à-canal). C'est d'ailleurs l'une des limitations de D-PROV que de ne pouvoir représenter que des workflows exécutables (MISSIER et al., 2013). Néanmoins, ces informations de provenance peuvent servir à détecter des écarts lors de leur exécution et corriger d'éventuelles anomalies (GARIJO et GIL, 2012). Les vastes travaux menés dans le cadre de wf4ever, visant à proposer des ontologies en lien avec les workflows (BELHAJJAME et al., 2018), contribuent également à enrichir les informations de provenances liées à l'exécution des workflows. La Figure 4.3 illustre comment cette provenance des résultats est modélisée dans l'ontologie *wfprov* (BELHAJJAME et al., 2013c).

Comparativement à D-PROV, wfprov adopte une démarche plus générique de modélisation de la provenance des résultats issus de l'exécution des workflows (MISSIER et al., 2013) en adoptant une démarche plus séquentielle. Ainsi, wfprov fait apparaître la notion d'étape de l'exécution des workflows par l'entremise de la classe *ProcessRun* [2]. Ces étapes sont regroupées dans des *WorkflowRun* qui décrivent alors l'activité d'"exécuter un workflow" dans son ensemble. Ces *ProcessRun* introduisent également la notion de dépendance avec les *Artifact* (*i.e.* les entités), modélisée par la propriété *usedInput*. Les *Artifact* qui sont les résultats produits par une étape de l'exécution du workflow y sont liés comme sorties par la propriété *wasOutputFrom*.

2. Nous avons omis le préfixe *wfprov :* pour faciliter la lecture.

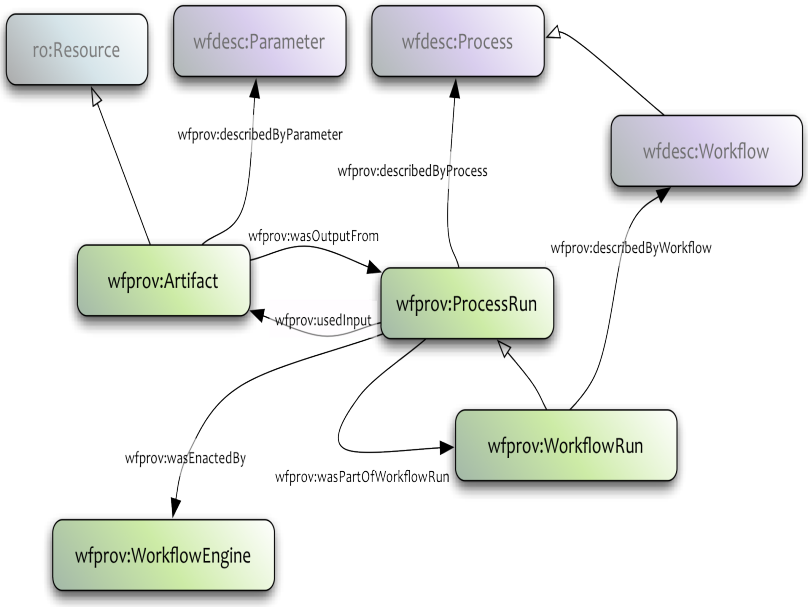


Figure 4.3.: Schéma de l'ontologie wfprov, intégrée à l'écosystème wf4ever (BELHAJJAME et al., 2013c). Les labels grisés font références à d'autres ontologies issues de wf4ever.

Une particularité de wfprov est aussi de pouvoir tenir compte du fait que différents *ProcessRun* peuvent avoir été réalisés dans des outils d'analyse différents. Chaque *ProcessRun* peut ainsi être mis en relation avec l'outil (*i.e. WorkflowEngine*) qui l'a mis en œuvre. Cette particularité vient corroborer le constat réalisé dans la Section 2.2 à propos de la diversité des outils pouvant être utilisés pour mener une analyse, et montre l'importance d'en tenir compte pour permettre un partage et une reproduction fiable de l'analyse (D. T. MICHAELIDES et al., 2016).

Néanmoins, apparaît une problématique directement liée à la représentation de l'exécution du workflow et son aspect technique. En effet, ce qui est capturé par de telles modélisations n'est pas le workflow en lui-même, ni la manière dont il est organisé, mais une instance exécutée de ce dernier, dans un contexte donné, qui se définit par les entités qu'il a alors produites et utilisées (GARIJO et GIL, 2011 ; MISSIER et al., 2013). De cette approche rétrospective résulte un impact négatif sur la compréhension et la réutilisation de la méthode générale mise en œuvre, autrement dit du workflow lui-même, puisqu'elle n'est pas partagée en complément de cette "modélisation rétrospective" (GARIJO et GIL, 2011 ; GARIJO et GIL, 2012 ; BELHAJJAME et al., 2018 ; D. T. MICHAELIDES et al., 2016).

Ainsi, des travaux proposent de modéliser comment les activités doivent être réalisées pour pallier ces manques de compréhension et de réutilisation, ce qui correspond à modéliser le plan des activités mises en œuvre. Toutefois, le plan d'une activité peut s'avérer complexe, faisant par exemple intervenir des notions d'allocation de ressources, de gestion des échecs ou même de méta-planification (OBERLE et al., 2006 ; GARIJO et GIL, 2012). Ainsi, même si D-PROV et, dans une moindre mesure, PROV *via* sa classe *prov :plan*, permettent de modéliser des informations de provenances qualifiables de prospectives (*i.e.* concernant le workflow lui-même), ils ne permettent pas nativement de capturer cette complexité inhérente à la mise en œuvre d'activités. Des informations peuvent alors manquer pour répliquer correctement une analyse, comme la configuration des activités ou leur ordre d'application.

P-Plan, en revanche, est un travail dédié à la représentation de la mise en œuvre des activités (GARIJO et GIL, 2012). Il s'agît d'une ontologie OWL développée pour décrire les workflows scientifiques sous forme de plans et pouvoir les relier aux instances de ces workflows. L'objectif de P-Plan est ainsi de donner la possibilité de publier ces plans d'exécution en même temps que les informations de provenances, pour favoriser la réutilisation, la qualité et la crédibilité scientifique des workflows. La Figure 4.4 schématise l'ontologie P-Plan.

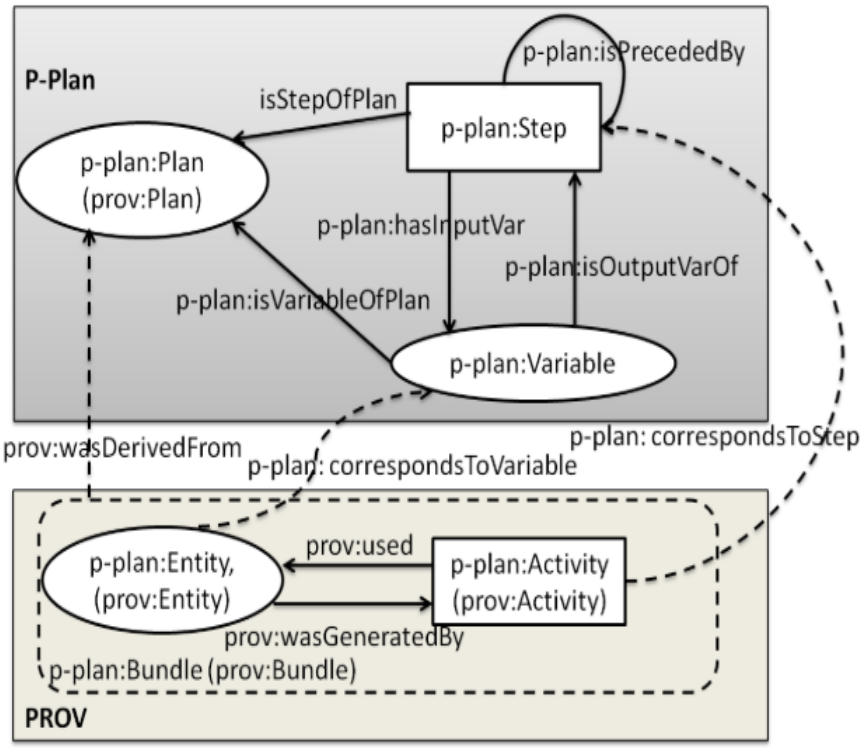


Figure 4.4.: Schéma de Ontology for Provenance and Plans (P-Plan) (GARIJO et GIL, 2012, p. 3).

Ainsi, P-Plan définit clairement la notion d'étape : une activité P-Plan, qui est subsumée par une activité au sens PROV, est décrite par des étapes (*p-plan :Step*) rattachées à un plan particulier. Ces étapes du plan modélisent les classes d'actions à appliquer lors de l'exécution d'une activité : entendez qu'il ne s'agît là pas de l'instance d'une méthode spécifique. En conséquence, cela apporte une abstraction supplémentaire au plan et le rend exécutable dans différents outils.

Notons ici que P-Plan introduit également une relation d'ordre entre les étapes d'une activité *via* la propriété *p-plan :isPreceededBy*. De cette manière, la transitivité et l'ordonnancement des étapes est capturée, ce qui permet de s'assurer de la bonne exécution d'un workflow. De plus, la description des variables d'entrées nécessaires aux étapes sont modélisées (*p-plan :Variables*), ainsi que celle des variables de sorties – pouvant potentiellement être les variables d'entrées d'une autre étape. Ces variables sont également associées à un plan particulier, et peuvent posséder des propriétés, comme un type, des métadonnées ou encore des restrictions.

L'un des avantages de P-Plan est d'étendre le vocabulaire PROV, tout en mettant en relation les nouveaux termes que cette ontologie propose avec ceux du standard. Ainsi, P-Plan utilise les classes et les propriétés natives de PROV pour décrire l'interaction des agents sur les plans (puisqu'un plan est, rappelons le, une activité PROV). Cela apporte une interopérabilité supplémentaire et une possibilité accrue de réutiliser les plans (GARIJO et GIL, 2012). Un autre avantage réside dans son approche générique qui rend possible des modélisations variées, en plus des workflows. Puisque P-Plan modélise les plans suivant un graphe acyclique des étapes et des entités qu'elles génèrent, il est par exemple possible d'y représenter les différentes étapes d'un protocole expérimental.

En revanche, à l'instar de PROV, cette généricité ne permet pas de capturer tous les éléments propres à un processus d'analyse sans introduire d'ambiguïté (MISSIER et al., 2013), comme les configurations d'une étape. Pour tenter de répondre à ce besoin, l'on voit émerger la notion de *workflow template* et des ontologies associées (GARIJO et GIL, 2011 ; GAJIRO et GIL, 2010 ; BELHAJJAME et al., 2018). Il s'agit, à la manière des plans, de fournir une représentation abstraite et non instanciée du processus à exécuter ; la différence majeure d'un template de workflow est de s'inscrire dans un écosystème sémantique dédié à la modélisation de ces processus, qui fait par exemple intervenir des termes

permettant de décrire le type de paramètres nécessaires à une opération. En outre, les ontologies capturant ces templates permettent communément de modéliser leurs instances ; en résulte ainsi une dualité entre workflows abstraits et workflows exécutables (GAJIRO et GIL, 2010).

Parmi ces travaux, l'on peut citer l'ontologie Open Provenance Model for Workflow (OPMW) (GAJIRO et GIL, 2010). Plus complexe que P-Plan ou D-PROV, elle permet cependant de décrire à la fois les traces d'exécution d'un workflow dans différents outils ainsi que sa représentation abstraite (*i.e.* son template). L'un des avantages d'OPMW est qu'elle étend P-Plan et PROV, par l'intermédiaire de l'ontologie OPM, qui est une autre ontologie utilisée pour modéliser la provenance des données (MOREAU et al., 2008 ; MOREAU et al., 2011), ce qui lui permet de renforcer les perspectives d'interopérabilité et de réutilisation des éléments modélisés.

OPMW spécialise ainsi la notion d'étapes (*i.e. p-plan :Step*), et structure le template sous la forme d'un enchaînement non plus d'étapes mais de "processus", qui décrivent le type de méthodes à suivre. OPMW permet également de définir quels types d'artefacts sont utilisés ou générés par une opération du template, tout en opérant sur ces artefacts un distinguo entre variables et paramètres du workflow. Il est aussi possible d'y spécifier d'éventuelles restrictions concernants ces artefacts.

Concernant la trace d'exécution des workflows, OPMW étend à la fois P-Plan, OPM et PROV. En réutilisant le vocabulaire de ces ontologies, outre favoriser l'interopérabilité et la réutilisation des éléments qu'elle modélise, OPMW permet de spécialiser un template de workflow pour en modéliser ses différentes exécutions, tout en spécifiant les liens qui existent entre eux. Par exemple, la notion d'artefact susmentionnée se voit subsumée par la notion d'entité de PROV (et d'artefact dans OPM), ce qui permet de représenter les instances des artefacts qui sont générées lors de l'exécution d'un processus spécifique.

Nous terminons cette section en présentant l'ontologie wfdesc contenue dans wf4ever (BELHAJJAME et al., 2018 ; PAGE et al., 2012). Elle est conçue pour décrire la *structure abstraite d'un workflow* (BELHAJJAME et al., 2013b), à savoir son template. À la différence d'OPMW, wfdesc n'étend pas directement des standards ontologiques comme PROV : il est toutefois possible de l'aligner avec ces standards (BELHAJJAME et al., 2013d). À la place, cette ontologie définit son propre vocabulaire et propose une modélisation plus riche des templates qu'OPMW – néanmoins plus spécialisée, en cela que wfdesc est le résultat de travaux tentant de répondre à des besoins terrains forts (K. M. HETTNE et al., 2012 ; BELHAJJAME et al., 2012b ; PAGE et al., 2012) – provenant majoritairement de Taverna (OINN et al., 2004) et *my*Experiment (C. A. GOBLE et D. C. DE ROURE, 2007). La Figure 4.5 schématise wfdesc.

Au niveau le plus abstrait de l'ontologie, le template est défini par *Workflow*. Cette classe est composée de *Process* [3], qui représente la classe des actions qui, quand exécutées, donnent lieu à l'opération qu'elles décrivent : ce sont donc des opérations non instanciées. Outre la possibilité d'imbriquer les workflows ensemble comme le montre la propriété *hasSubWorkflow*, l'une des particularités qu'introduit wfdesc est de considérer un template de workflow comme une opération non instanciée en le subsumant par *Process*.

Il s'agit ici d'un changement majeur de perspective concernant le template d'un workflow, puisqu'il n'est plus question de le considérer comme un artefact particulier, voire antinomique, aux opérations. Concrètement, ce changement permet de considérer les templates comme une ressource computationnelle instanciable qui est partageable et réutilisable dans d'autres analyses, notamment car ils possèdent eux aussi des paramètres et des variables spécifiques. Par extension, les opérations sont également devenues des opérations composites. Néanmoins, il faut noter que cette ontologie ne permet pas de classifier, ni de spécifier la nature de ces opérations (et donc, des workflows) (BELHAJJAME et al., 2013b).

Chacune de ces opérations (*i.e. Process* et *Workflow*) possèdent donc des variables d'entrée et des variables de sortie. Cependant, puisqu'il ne s'agit pas, dans ce cas, d'instances particulières – par exemple les variables d'entrée peuvent être assimilées aux paramètres d'entrée d'une fonction, wfdesc modélise ces variables comme des paramètres spécifiques en les subsumant par la classe *Parameter*. De

3. Nous avons omis le préfixe *wfdesc :* pour faciliter la lecture.

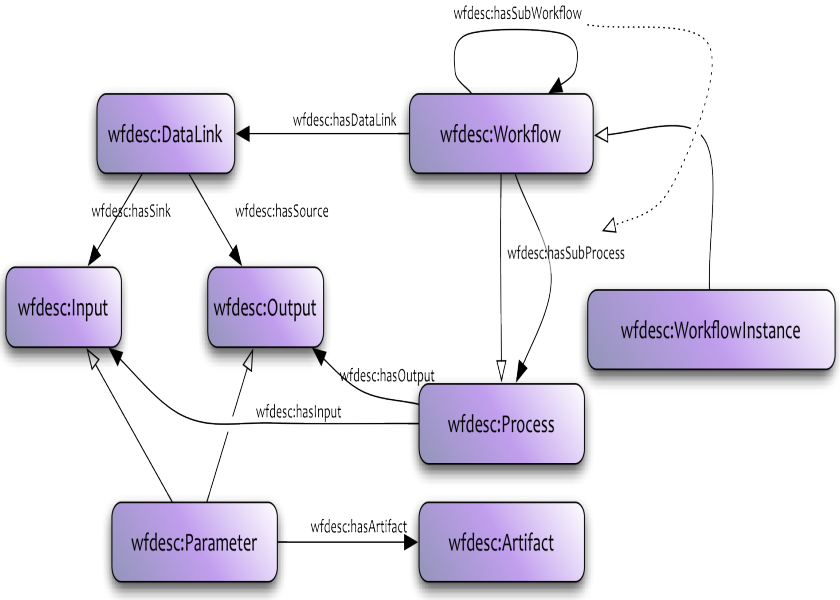


Figure 4.5.: Schéma de l'ontologie wfdesc, issue de l'écosystème sémantique dédié aux workflows wf4ever (BELHAJJAME et al., 2013b).

ce fait, il est également possible d'exprimer les configurations d'une opération qui ne relèvent pas des variables d'entrée (*e.g.* la manière de chercher les plus proches voisins).

Une autre particularité de wfdesc est de considérer à la fois les entrées, les sorties, mais également les paramètres comme des artefacts appartenant aux workflows. Ainsi, la notion d'artefact permet d'en définir son type syntaxique et sémantique, alors que la notion de paramètre définit son rôle vis-à-vis d'une opération précise. Cette ontologie ajoute également la notion de *DataLink*, approchant la notion de connexion entre deux tâches proposées dans D-PROV (MISSIER et al., 2013). Un *DataLink* est utilisé pour représenter la dépendance de deux opérations *Process* successives d'un workflow entre elles : cette classe indique quelles sorties (*i.e. Output*) sont nécessaires en entrée (*i.e. Input*) d'une autre opération.

Finalement, wfdesc permet de représenter une spécialisation du template de workflow *via Workflo-wInstance*. Il s'agit de définir toutes les configurations, les paramètres et les données nécessaires pour définir l'exécution du workflow. Cette classe s'explique en considérant wf4ever comme un écosystème sémantique pour les workflows. Elle permet de faire le lien entre la partie abstraite du workflow (*i.e.* wfdesc) et la partie instanciée (*i.e.* wfprov susmentionné). Comme il est possible de le voir dans la Figure 4.3 avec les rectangles aux labels grisés (partie haute de la figure), les instances des workflows sont, à l'instar de OPMW, fortement mises en relation avec leur template. Par conséquent, *WorkflowInstance* sert en quelque sorte d'indication pour la bonne réalisation d'une instance d'un workflow dans un contexte technique précis.

En guise de conclusion, il faut toutefois noter que, bien que les travaux ci-dessus ont retenu notre attention pour modéliser les processus d'analyse de traces d'apprentissage, il existe bien d'autres travaux pertinents. C'est par exemple le cas de l'ontologie M3PO (HALLER et al., 2006) qui permet de modéliser sémantiquement la chorégraphie d'un workflow (*i.e.* l'enchaînement des tâches) ou de Stato (STATO, 2018[b]), qui est dédiée au domaine des statistiques et qui permet par exemple de modéliser des conditions d'application de tests statistiques.

Les travaux que nous avons présentés au cours de cette section proposent de renforcer la sémantique des processus d'analyse. Cela constitue une ressource non négligeable pour modéliser les processus d'analyse de traces d'apprentissange, en favorisant leur compréhension, leur partage et leur réutilisation. Néanmoins, les modélisations présentées sont principalement centrées sur l'aspect technique du

workflow, et peu, voire pas du tout, sur son aspect contextuel. OPMW ne se dote que de quelques métadonnées par exemple. Or, comme le montrent les travaux de Chandrasekaran & *al.* (CHANDRASEKARAN et JOSEPHSON, 1997), adopter une modélisation sémantique peut également servir à conditionner l'application et le choix de paramètres de manière non ambiguë, et *in fine*, favoriser l'adaptation du processus d'analyse (si tant est que l'information y est disponible) et sa réutilisation.

4.1.2 ... À la modélisation de nouvelles informations

Modéliser sémantiquement les workflows et, à plus forte raison, les processus d'analyse selon leurs propriétés techniques résulte d'un fort impératif d'efficacité (LUDÄSCHER et al., 2006 ; BRODARIC et GAHEGAN, 2010 ; MISSIER et al., 2013). Comme étudié dans la section ci-avant, les ontologies modélisant la provenance des données se focalisent sur des ressources et des aspects techniques principalement bas niveaux (*e.g.* l'accès à un service web) qui permettent la mise en œuvre des artefacts qu'elles représentent. Or, nous pouvons constater que de telles modélisations occultent le discours scientifique et les ressources scientifiques associées, comme les hypothèses testées par ces processus (BRODARIC et GAHEGAN, 2010).

Cela s'explique par le fait que ces ontologies de provenance ne capturent pas les connaissances d'un domaine en particulier, puisqu'elles sont conçues pour favoriser l'interopérabilité de ressources techniques spécifiques (*e.g.* les activités), comme c'est le cas avec PROV (LEBO et al., 2013) ou OPMW (GAJIRO et GIL, 2010). Comme le font remarquer Brodaric et Gahegan (BRODARIC et GAHEGAN, 2010), les ressources ainsi modélisées sont assimilables à des "boîtes noires" pour les scientifiques, et *a fortiori* pour les différents acteurs en général, où le domaine, le matériel scientifique utilisé (*e.g.* une publication) et les connaissances produites sont décrites de manière ambiguë. En outre, ces ontologies de provenance n'offrent pas une expressivité suffisante pour tenir compte de toutes ces informations. Autrement dit, le contexte dans lequel s'inscrit l'analyse n'est pas représentable.

Pour les différents acteurs (*e.g.* scientifique, pédagogique) concernés par les processus d'analyse de traces d'apprentissage, il en résulte alors un raisonnement et une appropriation du processus plutôt centré technique que scientifique. Or ce type d'approche ne garantit pas la bonne compréhension ou réutilisation de ces processus (BELHAJJAME et al., 2015). En effet, ces ontologies servent de socle technique aux différents systèmes pour l'exécution des workflows, mais elles ne permettent pas d'inférer comment les connaissances produites sont affectées par les actions réalisées, et pourquoi. De ce fait, l'utilité et la découverte de connaissances par les acteurs se retrouvent intimement liées à leur faculté de les induire à partir d'aspects techniques, plutôt qu'à comprendre intrinsèquement les ressources en question (BRODARIC et GAHEGAN, 2010).

Pour remédier à ce manque de contextualisation de l'analyse, différents travaux proposent d'enrichir les ontologies dédiées aux workflows. Partant du postulat que les métadonnées sont un autre aspect important des workflows (MOREAU et al., 2010 ; K. M. HETTNE et al., 2012), le *modus operandi* de ces travaux est initialement de capturer ces métadonnées, qui sont habituellement exploitées manuellement, et de les sémantiser (LUDÄSCHER et al., 2006). Par exemple, OPMW (GAJIRO et GIL, 2010) capture des métadonnées associées soit au template du workflow (*e.g.* une documentation, un auteur, une date de création) soit à une exécution de ce template (*e.g.* le statut final de l'exécution).

Néanmoins, ce mode opératoire se heurte à la superficialité des ressources qu'il est possible de capturer et au fossé sémantique (SMEULDERS et al., 2000) que ce type d'approche peut générer entre l'utilisateur et le système : il ne permet pas de prendre en compte l'inhérente complexité des ressources utilisées dans les analyses (LUDÄSCHER et al., 2006), comme les liens potentiels entre la configuration des opérateurs et le contexte de l'analyse. Ainsi, d'autres travaux (BECHHOFER et al., 2013 ; BRODARIC et GAHEGAN, 2010 ; CYGANIAK et al., 2010 ; BELHAJJAME et al., 2015) se proposent de capturer une sémantique plus élaborée des ressources utilisées dans un processus, et qui puisse être exploitée par la machine. L'objectif est ainsi de permettre en plus de raisonner en terme d'étapes scientifiques et d'objectifs, et d'en fournir les ressources nécessaires.

Il faut toutefois modérer ces propos en remarquant que capturer la sémantique des ressources utilisées s'avère être une tâche complexe dans beaucoup de disciplines scientifiques, par exemple en ce qui

concerne les données, les hypothèses effectuées ou encore les environnements dans lesquels des mesures ont été réalisées. Il l'est encore plus de rendre cette sémantique exploitable au mieux par la machine (Bowers et Ludäscher, 2005 ; Ludäscher et al., 2006).

Les premiers travaux que l'on peut citer et destinés à capturer une sémantique plus riche à propos des données utilisées sont SCOVO (Hausenblas et al., 2009) et sa combinaison avec SDMX (ISO, 2005) dans le cadre d'une approche orientée web sémantique (Cyganiak et al., 2010). SDMX (*Statistical data and metadata exchange*) est un modèle ISO qui concerne la manière de représenter les données statistiques, mais également la manière de définir leurs différentes dimensions et attributs, en plus des obersvations associées, pour favoriser leur échange. Il permet ainsi de modéliser à la fois des informations techniques, comme le type des données, et des métadonnées comme la catégorie à laquelle appartiennent certaines données.

Quant à SCOVO (*Statistical Core Vocabulary*) (Hausenblas et al., 2009), il s'agit d'une ontologie légère (*i.e.* qui contient peu de termes) destinée à exploiter les avantages du web sémantique pour renforcer la description des données statistiques et leur partage. SCOVO définit pour cela trois concepts de bases : celui de *Dataset*, qui fait office de container de données (*e.g.* un tableur) ; le concept de *Data item*, qui représente une donnée et qui appartient à un dataset (*e.g.* une cellule d'un tableur) ; le concept de *Dimension*, qui permet d'exprimer les qualités d'une donnée (*e.g.* la localisation, le prix). La particularité de SCOVO est d'opérer un distinguo entre la sémantique structurelle de la donnée statistique et la sémantique de son domaine.

Cyganiak & al. (Cyganiak et al., 2010) constatent néanmoins des limitations à SCOVO, comme l'impossibilité de créer des sous-groupes de données et de les annoter ou encore le manque de disctintion entre dimensions, attributs et mesures. Pour combler ces limitations, ils proposent ainsi d'aligner ontologiquement SDMX avec SCOVO et RDF : ils conservent de cette manière la particularité de SCOVO concernant la sémantique structurelle et la sémantique du domaine, tout en apportant une certaine richesse descriptive propre à SMDX. L'une des conséquences qui nous intéresse ici est qu'il devient possible d'exploiter ces informations sémantisées du domaine pour à la fois décrire les variables et contextualiser les données statistiques, *via* l'utilisation du web sémantique. Par exemple, il devient possible de contextualiser des données statistiques identifiées comme géographiquement situées, en les enrichissant avec des informations supplémentaires, comme le pourcentage d'urbanisation du milieu.

D'autres travaux se sont portés sur la manière de capturer le discours scientifique à l'aide d'ontologies, puisqu'elles contribuent à rendre les connaissances scientifiques plus explicites, à détecter les erreurs et à promouvoir la communication (Soldatova et King, 2006). Le domaine des sciences biomédicales offre en ce sens des propositions intéressantes, directement motivées par la nécessité de pouvoir comprendre, reproduire et adapter les resultats publiés (Soldatova et King, 2006 ; Garijo et Gil, 2012).

L'ontologie proposée dans le cadre du projet *Semantic Web Applications in Neuromedicine* (SWAN) (Ciccarese et al., 2008) s'inscrit dans cette démarche. SWAN est une ontologie [4] destinée à répondre à plusieurs préoccupations en lien avec les affirmations scientifiques, leurs contenus et à la manière de les expliciter. Par exemple, il s'agit de tenir compte du contexte des expérimentations ou des théories et de comment ces dernières ont été validées. Ses principaux objectifs sont ainsi de faciliter le discours et la collaboration par l'entremise du web, de favoriser la découverte de nouvelles pistes de recherches et aussi de souligner le manque de preuves, voire les inconsistances pouvant exister dans ces affirmations scientifiques.

Pour cela, SWAN modélise cette notion de discours scientifique et en identifie six éléments principaux (Ciccarese et al., 2008 ; Ciccarese et al., 2018). Tout d'abord, (1) les personnes, les groupes et autres organisations qui constituent des **entités réelles**, et où il est possible de modéliser leurs affiliations et interconnexions ; (2) les **éléments du discours** qui modélisent les affirmations, les questions et les hypothèses survenant dans un discours scientifique, ainsi que les relations (*e.g.* argumentation) entre eux ; (3) les **références bibliographiques et citations** ; (4) les **entités concernées** par le discours et les expérimentations : dans le cadre de SWAN, il s'agit principalement d'entités biologiques et leur

4. Par souci de clarté de lecture, nous appelons SWAN l'ontologie du projet *Semantic Web Applications in Neuromedicine*

sémantique est issue de services dédiés ; (5) des **tags** et des **qualificatifs** permettant d'ajouter de l'information, venant toujours d'un vocabulaire précis ; et (6) des **informations de provenances et de versions**. Ce dernier point fait par ailleurs écho avec les travaux présentés en Section 4.1.1, et permet à SWAN d'être intégré, du moins partiellement, dans des approches de provenance centrées données (Garijo et Gil, 2012).

Concrètement, avoir identifié ces éléments permet de discourir sur l'ensemble du processus scientifique réalisé grâce à des informations sémantisées. Il devient ainsi possible d'exprimer de manière structurée et interprétable par la machine des informations scientifiques en lien avec les données initiales, les étapes d'une expérimentation (*e.g.* les hypothèses qui les sous-tendent) et les données intermédiaires, ou encore les conclusions d'un travail scientifique. Par extension, SWAN modélise aussi les informations de publication des résultats, par exemple les auteurs d'une étude publiée dans un article précis. En outre, SWAN permet également de capturer le cycle de vie des entités biologiques manipulées et de le décrire. Il faut aussi noter sa possibilité de capturer un discours scientifique complexe par la mise en relation des différents éléments du discours, ce qui fait émerger les prémisses d'une modélisation du débat et des conflits scientifiques.

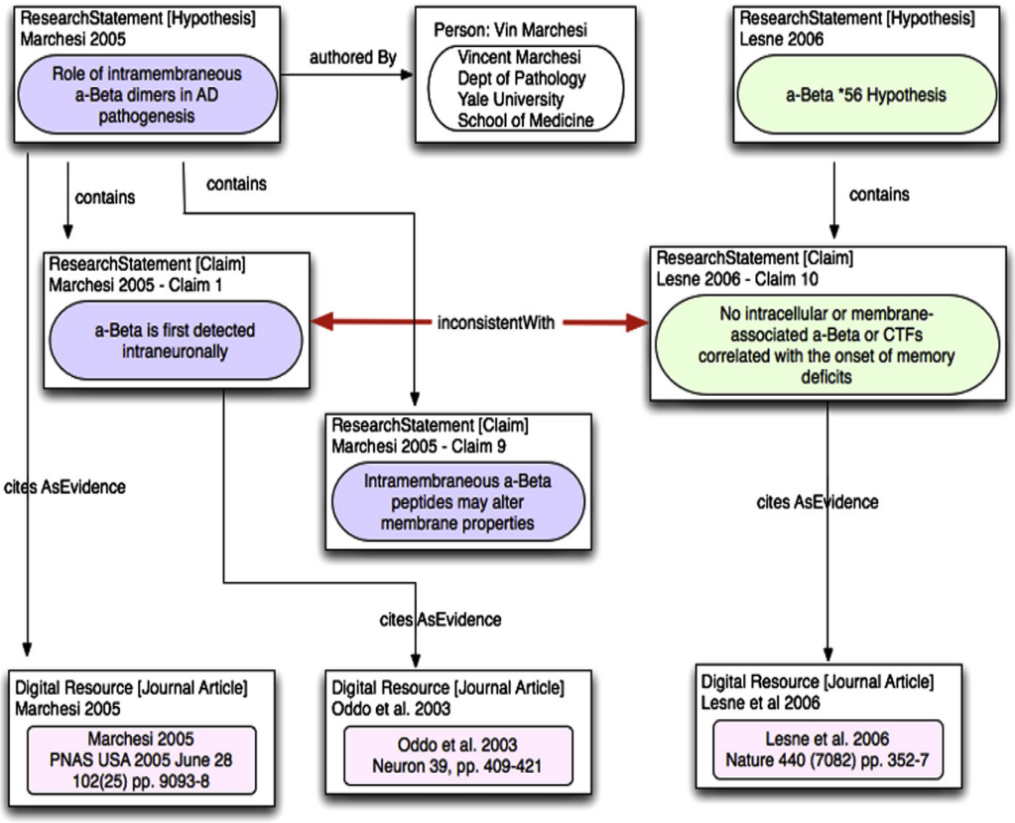


Figure 4.6.: Exemple d'instanciation de l'ontologie dans le cas d'une affirmation inconsistante dans le domaine de la biologie, qui fait intervenir un conflit entre hypothèses, preuves et affirmations (Ciccarese et al., 2008, p. 749).

La Figure 4.6 illustre en partie les possibilités d'une telle ontologie. Cet exemple illustre l'instanciation d'une sous-partie du vocabulaire de SWAN concernant la représentation d'hypothèses, d'affirmations et de preuves (Ciccarese et al., 2018). Contextuellement, il s'agit dans cet exemple de deux scientifiques en science biomédicale, Marchesi en violet à gauche et Lesne en vert à droite, qui tentent de développer des modèles explicatifs liés à la maladie d'Alzheimer. Pour cela, ils proposent des hypothèses dont celles présentées dans la partie supérieure de la Figure 4.6.

Outre la possibilité d'indiquer des informations liées à la paternité d'une proposition (cf. la relation *authoredBy* liée à l'hypothèse de Marchesi), la Figure 4.6 présente comment des entités scientifiques reposent sur d'autres et interagissent. Ici, l'hypothèse posée par Marchesi est constituée d'un ensemble

d'affirmations (*i.e.* notés *Claim*), et est appuyée par un article de recherche publié dans un journal, tout comme l'hypothèse proposée par Lesne. Notons par ailleurs qu'une hypothèse dans un contexte peut être réutilisée en tant qu'affirmation dans un autre (Ciccarese et al., 2008). Néanmoins, l'une des affirmations de Marchesi entre en conflit avec celle de Lesne, ce qui est ici matérialisé par une relation d'inconsistance entre elles (*i.e. inconsistentWith* en rouge). Ainsi, bien qu'il ne s'agisse pas d'une réfutation immédiate de l'une ou l'autre des hypothèses, SWAN permet de modéliser la présence d'un potentiel conflit et offre un cadre sémantique pour raisonner dessus.

Cependant, dans le cadre des workflows et, à plus forte raison, des processus d'analyse, SWAN ne permet pas de structurer l'intégralité de leurs informations de manière non ambiguë. En effet, il s'agit d'une ontologie qui n'intègre pas la technicité des tâches dans sa modélisation : il est difficile de modéliser directement les opérations réalisées et plus encore d'en modéliser le discours qui les concerne (Garijo et Gil, 2012). Certains travaux pallient ce manque en proposant des termes supplémentaires pour représenter cette technicité. Par exemple, l'ontologie OBI, pour *Ontology for Biomedical Investigations* (Bandrowski et al., 2016), se compose de plus de deux milles cinq cents termes (Bandrowski et al., 2018) couvrant à la fois des éléments du discours scientifique, de la méthode et de l'analyse réalisée.

Ainsi, de si volumineuses ontologies qui tiennent lieu de banques sémantiques ne capturent pas les relations qui peuvent exister dans les processus d'analyse, à la différence de travaux plus spécialisés (Garijo et Gil, 2011 ; Missier et al., 2013). Enfin, que ce soit ces travaux proposant un vocabulaire étoffé ou ceux destinés à modéliser le discours scientifique, ils ne permettent que de décrire les ressources supplémentaires qui sont utilisées lors de la démarche, par exemple pointer vers un jeu de données particulier. Ils ne permettent pas de les intégrer directement dans le processus d'analyse, alors que ces ressources œuvrent pour l'adaptation et la réutilisation des processus (Belhajjame et al., 2015).

Il est en effet important de pouvoir intégrer correctement les ressources – dont le discours scientifique – dans les processus d'analyse. Comme le font remarquer De Roure & *al.* (D. De Roure et al., 2010), les acteurs de l'analyse (*i.e.* scientifiques) exploitent une grande variété de données et de ressources, souvent hétérogènes. Concrètement, ces ressources sont des artefacts pertinents pour une tâche réalisée. Par exemple, une hypothèse est considérée comme une ressource, au même titre qu'une citation, un exécutable, un jeu de données ou des slides (Bechhofer et al., 2013). Toutes ces ressources, bien qu'individuellement utiles, ne supportent et contribuent à vraiment enrichir toute la démarche qu'une fois considérées dans leur globalité (D. De Roure et al., 2010).

La difficulté ici est de proposer un moyen de capturer la diversité de ces ressources et de les intégrer dans la structure même de l'analyse. L'objectif est important, puisque cela permet de renforcer la compréhension des analyses et, à terme, de s'assurer de leur qualité et de leur réutilisation (Belhajjame et al., 2012b). Pour conclure cette section, nous parlons des travaux concernant les *Research Objects* (Bechhofer et al., 2013) dans la discipline des workflows, destinés à tenir compte de ces ressources directement dans l'analyse.

Les *Research Objects* (RO) sont définis comme des agrégats sémantiquement enrichis de ressources qui forment des unités de connaissances, autrement dit un réseau d'informations essentielles à la compréhension intrinsèque des différents éléments de l'analyse et de la démarche réalisée. Cela concerne par exemple les données utilisées, les méthodes employées pour et analyser les données, mais également les personnes impliquées ou même le workflow en lui-même(Bechhofer et al., 2013).

D'un point de vue informatique, les RO sont des conteneurs partageables de ressources qui peuvent être produites et consommées par divers services et acteurs. En outre, ils supportent l'accès direct à ces ressources, ou *a minima* doivent permettre d'accéder à l'identification de ces ressources. Ils forment ainsi un cadre structurel qui permet de communiquer ces connaissances dans le cadre du web sémantique (Belhajjame et al., 2012a ; Belhajjame et al., 2012b). La Figure 4.7 présente l'ontologie qui permet de modéliser les RO. Cette ontologie s'inscrit dans la suite ontologique wf4ever, afin de lutter notamment contre les problèmes de réutilisation des workflows (Belhajjame et al., 2015). À ce titre, l'ontologie des RO est mise en relation avec les ontologies de template de workflows (*i.e. wfdesc*)

et de provenance (*i.e. wfprov*), comme le montre la Figure 4.3 de la section précédente (cf. partie haute gauche).

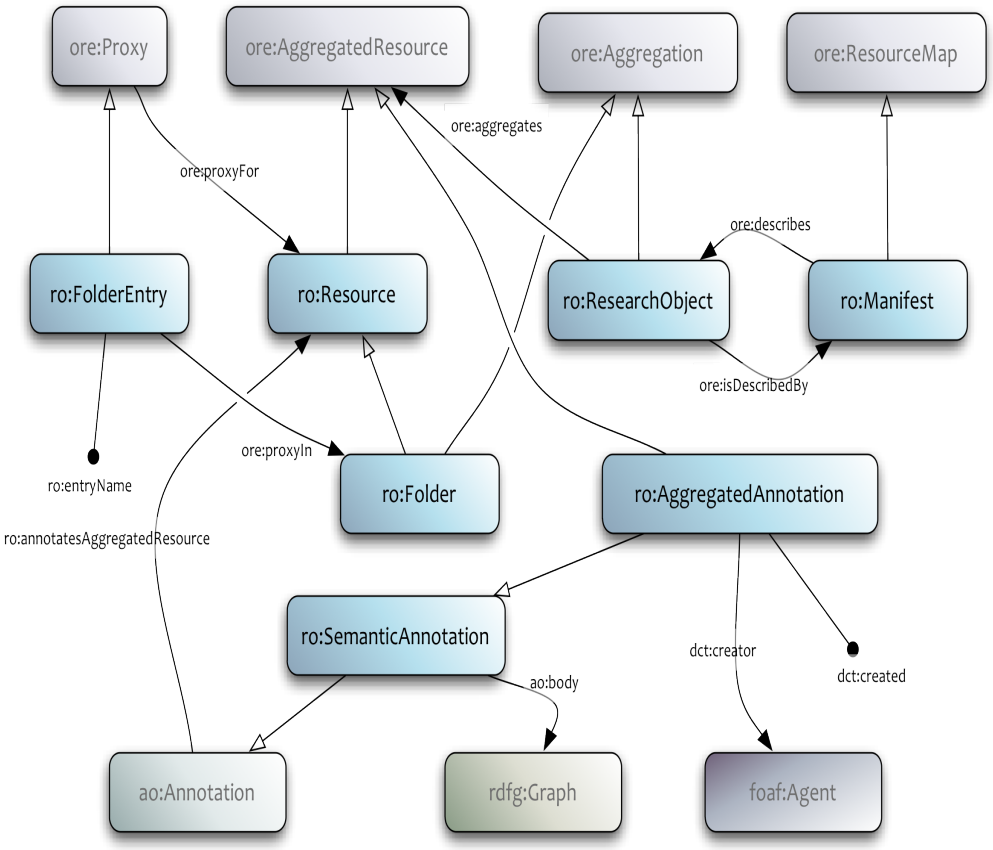


Figure 4.7.: Schéma de l'ontologie des Research Objects, issue de l'écosystème sémantique dédié aux workflows wf4ever (BELHAJJAME et al., 2013a)

Comme illustré dans la Figure 4.7, l'ontologie des RO étend (BELHAJJAME et al., 2012a) l'ontologie Open Archives Initiative Object Reuse and Exchange (OAI-ORE) (LAGOZE et al., 2008). Brièvement, cette dernière définie des standards pour décrire et échanger des agrégats de ressources web : elle définit donc les notions d'agrégat, de ressources et d'appartenance à un agrégat qui sont étendues dans RO. Ainsi, un RO est défini comme étant une agrégation (*i.e.* est subsumé par *ore :Aggregation*) de ressources – subsumées par le concept de ressources agrégées *ore :AggregatedResource*. Une ressource pouvant être n'importe quel artefact numérique.

Deux particularités sont à noter dans cette modélisation des RO. Tout d'abord, l'ontologie fortifie l'organisation structurelle des ressources et permet d'en capturer la hiérarchie qui peut exister entre elles. Pour cela, cette ontologie définit la notion de dossiers (*i.e. ro :Folder*), assimilable à l'organisation en dossiers des systèmes d'exploitation actuels. Il s'agit ici d'un type spécial d'agrégation, où chaque ressource agrégée y est explicitement associée (*i.e. ro :FolderEntry*). C'est particulièrement nécessaire afin de contrôler la portée des assertions liées à ces ressources lors des inférences du système. L'on note aussi qu'un dossier est également subsumé par la notion de ressources agrégées, ce qui lui permet à la fois d'être contenu dans des RO mais aussi de contenir d'autres dossiers.

La deuxième particularité de cette ontologie est de faire intervenir un mécanisme d'annotation spécifique. Les annotations (*i.e. ro :AggregatedAnnotation*) sont utilisées pour décrire sémantiquement les éléments constitutifs d'un RO, et le RO lui-même. À cet égard, ces annotations sont naturellement subsumées par le concept de ressources agrégées, mais également par celui d'annotation sémantique (*i.e. ro :SemanticAnnotation*). Puisqu'une annotation sémantique est associée à un graphe RDF (*i.e.*

rdfg :Graph), les annotations utilisées en lien avec les ressources du RO contribuent à y définir des éléments sémantiques supplémentaires, exploitables par la machine.

Aussi, cette approche des RO contribue au partage de ressources très diverses, nonobstant des possibilités de modélisation du discours et de l'investigation scientifique moindres, comparé à SWAN par exemple (CICCARESE et al., 2018). Combiné avec l'ontologie de provenance *wfprov* et de description des templates de workflows *wfdesc*, cette ontologie des RO est dans la pratique grandement utilisée (BELHAJJAME et al., 2012b ; BELHAJJAME et al., 2013d ; BÁNÁTI et al., 2015). C'est par exemple le cas de ^{my}Experiment et de Taverna, qui favorisent le stockage et le partage des ressources utilisées lors de la mise en œuvre des workflows, ainsi que les workflows eux-mêmes (C. A. GOBLE et D. C. DE ROURE, 2007).

Par ailleurs, cette appropriation du concept de RO dans le monde biomédical a permis à Becchofer & *al.* (BECHHOFER et al., 2013), à la suite d'une étude empirique, d'identifier sept stéréotypes d'unités de connaissances couramment véhiculés dans les RO. Outre les connaissances en lien avec la publication au sens large, les RO sont également utilisables pour capturer le contexte des données – alors appelés View/Context Object. Plus atypique, ils se prêtent à la représentation de travaux en cours d'élaboration, où potentiellement une grande quantité d'acteurs peuvent intervenir ; mais ils servent aussi à archiver les workflows et leurs ressources une fois terminés.

Pour conclure, toutes ces propositions permettent d'enrichir les processus d'analyse avec des éléments différents, dans une démarche sémantique. Il est ainsi vraiment intéressant de noter qu'une proposition comme les RO, plutôt générique dans sa démarche, permet déjà l'émergence de comportements plus complexes dans la modélisation des analyses, de la part des différents acteurs. De plus, ces travaux semblent supporter une approche itérative et interactive de l'analyse (BELHAJJAME et al., 2012a). Aussi, ces approches apportent des pistes solides pour le partage des analyses, mais également pour favoriser leur compréhension et leur réexploitation.

Néanmoins, une problématique qui réside dans la modélisation sémantique, outre la modélisation elle-même, est la manière dont les réseaux sémantiques sont exploitables pour bénéficier aux différents acteurs et pas uniquement à la machine. En effet, en plus des possibilités de raisonnements (*e.g.* inférences) que ces réseaux peuvent ou ne peuvent pas fournir à la machine, les acteurs y sont également confrontés plus ou moins directement. Il est donc nécessaire que ces acteurs puissent eux aussi utiliser ces réseaux sémantiques efficacement, entre autre afin de prévenir un fossé sémantique trop important.

4.2 Interpréter et exploiter la sémantique à travers les requêtes utilisateurs

Dans un cadre informatique, il est manifeste qu'avoir accès à des processus d'analyse de traces d'apprentissage qui puissent être, une fois partagés, adaptables et réutilisables induit la nécessité de pouvoir répondre aux requêtes des différents acteurs lorsqu'ils veulent y accéder et s'en servir. Autrement, la capitalisation serait inexistante, puisqu'il ne s'agirait alors que d'un dépôt, sans possibilité aucune de chercher, de trouver, ni de récupérer, un processus ou un opérateur quelconque. Dès lors, cela implique deux conditions : que de tels processus d'analyse puissent être interprétables par la machine ; et que les requêtes des acteurs envers le système puissent également être interprétables, c'est-à-dire projetées dans le domaine sémantique de la modélisation utilisée pour les processus d'analyse.

Mais il faut noter que les capacités de raisonnements et d'inférences, constitutives de l'interprétabilité dont nous parlons, sont directement dépendantes des modélisations adoptées (VAN HARMELEN et al., 2008). Et, comme nous l'avons vu dans les sections précédentes, il en existe une grande variété, soit pour les processus d'analyse, soit pour les workflows ou des artefacts plus généralistes : modèles, méta-modèles ou encore ontologies. La manière d'interpréter les processus d'analyse, tout comme les requêtes utilisateurs et la manière d'y répondre, revêtent donc des solutions *ad-hoc* qui sont propres à

chacun des outils d'analyse et potentiellement conditionnées par des prérequis computationnels (cf. Section 2.2).

Dans cette section, nous revenons dans un premier temps sur certains des travaux de modélisation et d'outils d'analyse déjà présentés. Nous explorons comment ces outils d'analyse prennent en charge les requêtes utilisateurs et, lorsque les sources scientifiques ou techniques sont suffisantes pour s'en assurer, comment ils les traitent pour rechercher des résultats adéquats. Dans un second temps, nous nous intéressons aux travaux issus du web sémantique concernant l'approximation des requêtes utilisateurs, qui apportent des pistes pertinentes pour tenir compte de la complexité inhérente aux processus d'analyse et aux traces dans un cadre sémantique.

4.2.1 Une étude des capacités des outils d'analyse actuels

Étudier les capacités à répondre aux requêtes utilisateurs et à rechercher des résultats dans les outils d'analyse nous amène premièrement à considérer pourquoi ces outils ont été conçus. Il paraît en effet naturel qu'un outil destiné à favoriser le partage des processus d'analyse de traces, comme UnderTracks (Mandran et al., 2013), qui s'établit dans un contexte communautaire plus riche, permette et traite des requêtes utilisateurs de manières différentes d'un outil d'analyse plus classique, comme RapidMiner (Hofmann et Klinkenberg, 2013).

Toutefois, c'est ce contraste qu'il est intéressant d'observer. Il fait surgir de nouvelles nécessités directement liées à la capitalisation, plus l'outil tend vers cette propriété. Par exemple, il existe une différence notable entre SPSS et Galaxy – outils que nous abordons par la suite – sur la manière de gérer les requêtes utilisateurs, les éléments qui sont utilisés pour y répondre et par quels moyens ces éléments sont interprétés par le système. Ce contraste nous apporte aussi des pistes sur la nécessité de réduire le fossé sémantique entre utilisateurs et systèmes, afin de favoriser la réutilisation des analyses (Belhajjame et al., 2013d ; Belhajjame et al., 2015).

Si nous étudions au préalable des outils d'analyse classiques et inter-domaines, nous remarquons que la liberté qu'ont les utilisateurs pour communiquer avec le système se retrouve restreinte à seulement quelques archétypes de requêtes. Par exemple, que ce soit dans RapidMiner ou Orange : Data Mining (Demšar et al., 2013), il n'est pas possible pour un utilisateur de chercher un opérateur à partir du type de donnée qu'il produit ni, à plus forte raison, de chercher un processus d'analyse d'après les connaissances qu'il produit.

Par contre, dans le cas de SPSS (IBM, 2018), les requêtes sont principalement centrées sur les données : les utilisateurs peuvent surtout requêter le système pour rechercher des données ou des variables spécifiques. Le contenu de ces requêtes est exprimé sous la forme de texte libre, et les termes sont ensuite comparés aux données préalablement chargées. Cela revient, pour l'outil, à répondre à la question "Trouver toutes les données dont le nom contient **X**". La recherche est modulable à l'aide de certains paramètres, comme la prise en compte ou non de la casse, ou encore une correspondance exacte entre tous les termes. Dès lors, ces requêtes ne requièrent pas de l'outil d'être capable d'interpréter les données (*e.g.* inférer le type d'une donnée). Il en résulte une non-exploitation des relations qui peuvent exister entre les données, outre l'impossibilité pour les utilisateurs d'interroger l'outil pour des propositions plus complexes, comme la recherche de commentaires, d'opérateurs ou de leurs configurations.

D'autres outils inter-domaines, comme RapidMiner, complémentent l'espace des requêtes disponibles pour l'utilisateur (RapidMiner, 2018[b]). Pour rappel, RapidMiner est un outil d'analyse utilisant le paradigme des workflows pour représenter l'analyse. L'abstraction alors accrue du flux d'opérations dans ce type de modélisation (cf. Section 2.2) offre un terrain propice à l'ajout de nouveaux éléments descriptifs. L'on remarque par exemple dans cet outil l'utilisation de tags qui sont associés à divers éléments, dont les opérateurs ; ce sont des courtes chaînes de caractères descriptifs.

Aussi, dans RapidMiner, les requêtes ne concernent plus que les données (RapidMiner, 2018[a]). Le système est également capable d'interroger les opérateurs, qu'ils soient natifs ou situés dans un stockage décentralisé (*e.g.* marketplace), pour vérifier s'ils correspondent aux requêtes utilisateurs. Au

même titre que SPSS, les requêtes sont exprimées sous forme de texte libre. L'outil réalise donc une comparaison terme à terme avec le nom des différents éléments. Toutefois, RapidMiner peut interroger des champs descriptifs supplémentaires, comme les tags susmentionnés. Dans une certaine mesure, et puisque ces champs rajoutent des qualificatifs supplémentaires, il devient alors possible pour l'outil de requêter les éléments selon certaines de leurs propriétés, si tant est qu'elles aient été au préalable correctement renseignées dans ces champs ajoutés manuellement.

Il faut enfin noter que les résultats des requêtes sont ici triés en fonction de leur date de dernière modification. Implicitement, cela revient à dire à l'utilisateur que le dernier élément modifié et qui contient les termes recherchés est le plus pertinent pour sa requête. Il s'agit évidemment d'une approche triviale, induite par le manque de relations et de sémantique à la fois dans la manière de décrire la requête et dans la modélisation des workflows (Belhajjame et al., 2012b). Ceci fait émerger la question de la pertinence d'un résultat vis-à-vis d'une requête donnée : c'est un problème complexe où le système doit être en mesure d'interpréter différents paramètres d'un élément pour lui attribuer un score (W. Bruce Croft, 2009). D'une manière générale, et on le remarquera aussi avec les travaux suivants, l'organisation et l'explication des résultats des requêtes utilisateurs – concernant les processus d'analyse – sont laissés au second plan. Or, ces propriétés contribuent à la compréhension et à la réutilisation des analyses (Volle, 2001 ; Koedinger et al., 2010 ; Kämpgen et Harth, 2011).

Dans UnderTracks, qui est concerné par le partage des processus d'analyse par l'intermédiaire d'un entrepôt commun, l'on voit apparaître des méthodes de recherche supplémentaires dédiées à soutenir cette propriété (Bouhineau et al., 2013b ; Mandran et al., 2013). En plus de permettre aux utilisateurs d'interroger l'entrepôt pour y chercher des traces ou des opérateurs spécifiques, il leur est aussi possible de le requêter à propos de processus d'analyse. La particularité ici est l'introduction de l'utilisation de métadonnées dans la recherche des différents éléments comme critères supplémentaires de pertinence. Il devient par exemple possible de rechercher par auteur d'une analyse. Néanmoins, à l'instar de SPSS ou encore de RapidMiner, le contenu de la requête est exprimée en texte libre ; puis une requête de comparaison de base de données relationnelle est exécutée pour récupérer les résultats, laissant supposer qu'il n'y a pas de raisonnement sur le contenu de l'entrepôt.

L'on observe, dans les travaux concernés par le partage et la réutilisation des analyses, cette tendance à exploiter des métadonnées pour supporter la recherche des utilisateurs. Par exemple, dans les premiers travaux de ^{my}Experiment (D. De Roure et al., 2010) ne faisant pas encore intervenir les ontologies de wf4ever (cf. Section 4.1.2), le système peut s'appuyer sur différentes métadonnées pour rechercher des workflows pertinents vis-à-vis de la requête utilisateur. Parmi celles-ci, et outre les tags, il y a des métadonnées concernant la qualité comme la notation des workflows ; le partage comme les permissions liées aux workflows ou leur nombre de vues ; ou encore les métadonnées en lien avec l'utilisation et le référencement des workflows.

Dans Galaxy (D. De Roure et al., 2010) – qui est un écosystème décentralisé de création, de modification et d'exécution de workflows, pour répondre aux requêtes utilisateurs, il est en plus possible d'utiliser les liens présents entre les différents éléments. En effet, lorsque des éléments sont publiés dans Galaxy, des liens sont alors créés pour en favoriser le partage et la réutilisation. En outre, il est également possible de rechercher dans l'historique des workflows. Néanmoins, toutes ces requêtes sont toujours exprimées avec du texte libre dans des champs de recherche, où certains filtres peuvent y être utilisés afin de les spécialiser.

Mais c'est dans les approches dotées d'un réseau sémantique que l'on constate d'importantes évolutions concernant les requêtes et leurs mises en œuvre (Missier et al., 2013 ; Belhajjame et al., 2013d ; Belhajjame et al., 2015). Ces approches fournissent un cadre propice au raisonnement puisqu'elles favorisent l'exploitation des relations sémantiques et taxinomiques qui peuvent exister entre les différents éléments constitutifs d'une analyse. L'interprétation des requêtes utilisateurs s'améliore également, puisque les termes utilisés lorsqu'ils les décrivent peuvent alors être alignés avec ceux du vocabulaire utilisé pour modéliser les analyses. Par exemple, indiquer qu'une chaîne de caractères est un *foaf :name* permet d'exploiter toutes les relations appartenant à ce terme du vocabulaire.

Avec l'introduction des *Research Objects* et des différentes ontologies issues de wf4ever (Belhajjame et al., 2018) implémentées dans Taverna et ^{my}Experiment, les utilisateurs ne sont plus restreints lors

des recherches aux informations superficielles des analyses. Il leur est possible d'interroger les outils pour récupérer des informations plus complexes (Belhajjame et al., 2015), grâce à la modélisation à la fois plus fine et sémantique qu'apportent ces travaux. Sur un plan technique, ces interrogations s'effectuent désormais en SPARQL, qui est un langage et un protocole de requête qui permet d'exploiter les graphes RDF.

Dans l'absolu, les possibilités de recherche que nous avons déjà évoquées sont couvertes par ces travaux. Il est par exemple possible de rechercher l'auteur d'un workflow, ou bien d'une ressource associée à ce workflow. Toutefois, comme le montre la Figure 4.8, elles sont également étendues en permettant des requêtes plus complexes, qui exploitent la modélisation sémantique. La partie haute de cette figure en est un exemple. Il s'agit d'une requête SPARQL destinée à trouver le ou les workflows qui ont été utilisées pour produire une annotation d'un gène spécifique (Belhajjame et al., 2015). Plus précisément, cette requête sert à identifier l'analyse qui a conduit à un résultat donné.

```
PREFIX
rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX prov: <http://www.w3.org/ns/prov#>
PREFIX wfprov: <http://purl.org/wf4ever/wfprov#>
PREFIX wfdesc: <http://purl.org/wf4ever/wfdesc#>
select distinct ?workflow ?def
where {?output wfprov:describedByParameter ?parameter ;
prov:wasGeneratedBy ?processrun .
filter regex(str(?parameter), "concept_ids") .
?processrun rdf:type ?type; wfprov:wasPartOfWorkflowRun
    [wfprov:describedByWorkflow ?workflow ].
?workflow wfdesc:hasWorkflowDefinition
    [<http://purl.org/pav/importedFrom> ?def] }
```

```
<?xml version="1.0"?>
<sparql xmlns="http://www.w3.org/2005/sparql-results#">
  <head>
    <variable name="workflow"/>
    <variable name="def"/>
  </head>
  <results>
    <result>
      <binding name="workflow">
        <uri>http://ns.taverna.org.uk/2010/workflowBundle/f7b17c46-a5c5-428f-ad45-a0b0f4e85935/workflow/
        Annotate_gene_list_w/</uri>
      </binding>
      <binding name="def">
        <uri>http://sandbox.wf4ever-project.org/rodl/ROs/data_interpretation/
        annotate_genes_biological_processes_xpath_cpids.t2flow</uri>
      </binding>
    </result>
    <result>
      <binding name="workflow">
        <uri>http://ns.taverna.org.uk/2010/workflowBundle/f7b17c46-a5c5-428f-ad45-a0b0f4e85935/workflow/
        Annotate_gene_list_w/</uri>
      </binding>
      <binding name="def">
        <uri>http://sandbox.wf4ever-project.org/rodl/ROs/data_interpretation-2/
        annotate_genes_biological_processes_xpath_cpids.t2flow</uri>
      </binding>
    </result>
```

Figure 4.8.: Illustration d'une requête utilisateur sur l'ontologie wf4ever et d'une partie des résultats obtenus une fois la requête exécutée. En haut, la requête, en bas les résultats partiels. D'après (Belhajjame et al., 2015).

Cette requête permet déjà d'envisager la modification du rapport de certains acteurs vis-à-vis de la consultation de l'existant. Ici, plutôt que de chercher des workflows à partir des données possédées, il devient alors possible de rechercher à partir de ce qui est attendu dans le cadre d'un besoin d'analyse. D'autres types de requêtes sont d'ailleurs possibles pour favoriser ce changement de comportement

dans la recherche. L'on peut citer par exemple la possibilité de trouver les entrées utilisées pour exécuter un workflow, ou encore identifier tous les workflows qui ont été modifiés entre deux versions d'un même élément.

Avant de conclure, et pour appuyer nos propos précédents sur les résultats, il est intéressant d'observer les résultats de la requête présentée dans la Figure 4.8 lorsqu'elle est évaluée dans Taverna. Un aperçu des premiers résultats est présenté sur la partie inférieure de la Figure 4.8. Il s'agit d'un fichier XML listant les workflows qui ont produit les données recherchées, passées en paramètre de la requête. Ce que l'on constate ici est l'abscence d'informations supplémentaires permettant d'identifier la raison et les éléments (*e.g.* des relations) contribuant à considérer ces workflows comme résultats, et leur pertinence. Il en reste ainsi une liste des processus d'analyse, laissés à l'appréciation des utilisateurs qui n'ont pas toujours conscience des mécanismes de requête déployés au sein des systèmes de recherche.

Au final, être capable de supporter les requêtes utilisateurs est important pour réellement permettre une capitalisation des processus d'analyse. Pour ce faire, il est important d'exploiter au maximum la modélisation des analyses et d'être capable de raisonner avec les éléments stockés. Toutefois, et puisque les utilisateurs peuvent ne pas avoir les mêmes connaissances, ni le même vocabulaire, la question d'une interprétation efficace par le système des requêtes utilisateurs permettant de limiter le fossé sémantique se pose. En effet, avec les solutions présentées, si un terme d'une recherche contient une phote d'aurtaugrafe il ne sera alors pas identifié correctement dans les éléments interrogés, amenant alors le système à générer un ensemble vide de résultat. De même, si l'utilisateur recherche au mauvais endroit des concepts qui sont pourtant similaires. Dans la section suivante, nous nous intéressons à ce problème, et à comment il est possible d'interpréter et d'approximer les requêtes utilisateurs.

4.2.2 Approximation des requêtes

D'une manière générale, modèles et ontologies sont réalisés par des spécialistes du domaine, qui en projettent alors l'utilisation qu'en feront les acteurs amenés à les utiliser. De cela, il résulte que ces acteurs peuvent ne pas comprendre entièrement, voire ne pas partager, certains concepts et certaines décisions prises lors de l'élaboration de ces modèles et ontologies : l'on parle alors de *mismatch* (Corby et al., 2006a). Cela est d'autant plus enclin à survenir dans les EIAH tant le domaine est riche par nature : différents acteurs (*e.g.* chercheur en didactique, en learning analytics, statisticien) aux expertises variées cohabitent pour renforcer la pédagogie dispensée dans des environnements hétérogènes à des apprenants qui, eux aussi, sont singuliers (Siemens, 2005). La conséquence directe est que, lorsque les acteurs interrogent le système, les résultats peuvent être inadéquats (*e.g.* un mauvais concept est utilisé par rapport à ce qu'avaient prévu les spécialistes), ce qui contribue à renforcer le fossé sémantique entre acteurs et systèmes (Smeulders et al., 2000 ; W. Bruce Croft, 2009 ; Corby et al., 2006a).

En outre, le domaine même des EIAH et sa richesse concourent à l'apparition de concepts et de connaissances difficilement modélisables, par exemple considérer qu'un apprenant est actif s'il se connecte souvent. Ces types de représentations mentales sont difficiles à formaliser, principalement parce qu'ils font intervenir des notions vagues, imprécises, voire incomplètes, que des modélisations classiques comme les ontologies définissent péniblement (Dubois et Prade, 1991). Dès lors, raisonner avec ces représentations dans des logiques de description (*DL*) – notamment utilisées pour définir les ontologies – s'avère complexe. En effet, le domaine d'interprétation de ces représentations s'étend au-delà de la sémantique bi-valuée supportée par les relations binaires de ce type de logique (Baader et al., 2003 ; Yen, 1991 ; Sanchez, 2006), ce qui empêche un système de raisonner correctement et d'utiliser les bons concepts et relations pour le faire.

Par exemple, prenons le cas d'un utilisateur qui interroge un système pour obtenir un processus d'analyse capable d'identifier les apprenants **actifs** dans un dispositif d'enseignement. Ce dernier est ouvert pendant une semaine aux apprenants. Le système exploitant alors une ontologie, le caractère vague du terme **actif** n'est pas pris en compte. Le système comparera sémantiquement le terme **actif** de la requête avec les individus qui peuplent l'ontologie, et identifiera les correspondances positives comme de potentiels résultats. Néanmoins, l'on peut facilement admettre que les processus résultats

d'une telle requête concernent des dispositifs pédagogiques différents de celui de l'utilisateur qui interroge le système, en particulier en ce qui concerne leur période d'accessibilité (*e.g.* un semestre, une année). Or, analyser l'activité des étudiants sur une échelle de temps si différente ne relève pas des mêmes techniques d'analyse. Cela peut conduire l'utilisateur à improprement répondre à son besoin avec des techniques inappropriées : des analyses identifiant des participations que l'on qualifierait de **régulière** (*e.g.* étude de la régularité de connexion de l'apprenant dans un intervalle de temps donné) ou de **neutre** (*e.g.* étude de l'activité par rapport à la moyenne des activités des apprenants dans le temps) pourraient elles aussi être de potentielles candidates.

Ces problématiques ont donc un rôle majeur dans la recherche d'information (*Information Retrieval*). Corby & *al.* (Corby et al., 2006a) font d'ailleurs remarquer à ce propos différentes conséquences. Un utilisateur peut être amené à ne pas utiliser les bons concepts du point de vue du créateur de l'ontologie, ce qui peut au final lui éclipser de potentiels résultats. De plus, si les concepts choisis ne peuvent pas exactement être mis en correspondances dans l'ontologie, alors le raisonnement échouera. Or, au lieu d'une réponse vide, un utilisateur pourrait préférer récupérer des résultats concernant des concepts similaires. Par exemple, une requête concernant la classification d'étudiants selon leur réussite qui n'aboutit à aucun résultat pourrait être étendue sur des concepts similaires de prédiction et d'échec, afin d'essayer de satisfaire la requête. Enfin, les propriétés relationnelles des concepts d'une ontologie ne sont pas toujours connues des utilisateurs lorsqu'ils interrogent le système. Il est donc important de leur permettre de récupérer des éléments précis indépendamment de la complexité relationnelle qui peut exister entre ces éléments.

Les efforts en lien avec l'approximation sémantique des requêtes constituent une piste intéressante pour traiter ces problématiques. Plus particulièrement, les travaux réalisés dans le cadre de Corese (Corby et al., 2006b) proposent des solutions concrètes pour répondre au problème de *mismatch* entre acteurs et concepteurs d'ontologies. Pour cela, le système gère les requêtes qui n'ont pas de résultats exacts en approximant soit la sémantique de la requête utilisateur, soit sa structure, voire les deux.

Pour présenter brièvement Corese, il s'agit d'un moteur de recherche destiné à requêter le web sémantique (Corby et al., 2004). Une particularité de ce moteur est de fonctionner avec une représentation interne de l'information, sous forme de graphes conceptuels (CG) (Sowa, 1984). Ainsi, lorsque Corese répond à une requête annotée dans une ontologie donnée (exprimé en RDFS ou OWL Lite), il procède à une projection de ces éléments vers un modèle de graphe conceptuel associé. Cette projection, qui est interprétable comme une fonction surjective, conserve les différentes propriétés qui peuvent être présentes dans l'ontologie, comme celle de subsomption. Dès lors, l'évaluation s'effectue en projetant le CG de la requête (CG^R) dans le CG de l'ontologie (CG^O) en question. Sommairement, cette évaluation repose sur l'implication logique de CG^R dans CG^O où CG^O est une spécialisation de CG^R, autrement dit que chaque concept de CG^R peut être projeté sur un nœud de CG^O, à la condition qu'il en possède le type.

Pour approximer les requêtes, les auteurs ont enrichi Corese avec quatre techniques – complémentaires – qui peuvent être réparties en deux catégories : trois en lien avec l'approximation ontologique, l'autre en lien avec l'approximation structurelle (Corby et al., 2006b ; Corby et al., 2006a). Parmi l'approximation ontologique, l'on trouve d'abord l'évaluation de la distance ontologique (*ontological distance*). La distance ontologique ici sert à définir le degré de filiation des différents termes entre eux, en s'appuyant sur la structure de l'ontologie. Elle s'inspire du calcul des distances dans un graphe conceptuel, ce qui permet de réaliser des projections non binaires. L'intuition mise en avant ici est que plus les termes issus d'un même taxon sont situés en profondeur dans l'arbre taxinomique, plus ils sont proches.

En second lieu, Corese introduit la notion de proximité contextuelle (*contextual closeness*). Il s'agit de tenir compte des liens sémantiques qui peuvent exister entre concepts, mais qui ne sont pas capturés par une relation de subsomption. L'objectif est de pouvoir identifier et représenter une certaine proximité entre de tels concepts. Cette proximité a pour effet de réduire la distance ontologique entre les concepts recherchés (potentiellement à une notion de nœuds frères), ce qui a pour conséquence de renforcer l'approximation. Pour modéliser cette proximité contextuelle, Corese utilise la propriété *rdfs :seeAlso* du standard RDFS, pour lier les concepts entre eux. Elle est définie comme taxinomiquement transitive, si l'on peut le dire ainsi : un concept qui possède une approximation contextuelle vers un deuxième

concept, et qui est subsumé par un tout autre concept, voit ce dernier également approximé à ce deuxième concept.

Dernier élément en lien avec l'approximation ontologique : la projection approximative (*approximate projection*). Cette projection consiste à interroger le système à partir d'une requête exacte, puis avec différentes requêtes approximant celle initiale. Pour ce faire, ces requêtes reposent sur les techniques de distance ontologique et de proximité contextuelle. De cette manière, il est possible d'obtenir des résultats qui ne sont pas uniquement subsumés par les concepts contenus dans les requêtes, mais qui sont suffisamment proches sémantiquement. De là, la pertinence des résultats est mesurée par rapport à leur similarité vis-à-vis de la requête. Cette similarité dépend principalement de la distance ontologique des différents concepts entre les requêtes approximées et leurs résultats [5]. Enfin, pour que les résultats soient jugés pertinents par le système – puisque le système sera potentiellement amené à réaliser des approximations trop importantes, leur valeur de similarité est comparée à un seuil, qui est établi relativement à la meilleure similarité qui a été calculée.

Enfin, l'approximation structurelle de Corese réside dans sa manière de traiter la structure de la requête avec celles des résultats. Lors de la recherche, le moteur est capable d'abstraire des relations intermédiaires qui pourraient se situer entre les concepts qui sont recherchés : l'utilisateur peut ainsi chercher des termes conceptuellement liés, le tout sans savoir comment ces termes sont en relation entre eux. Pour ce faire, Corese introduit la notion de *graphe de chemin* entre deux termes et de *longueur de chemin* dans son mécanisme de requêtage. Un graphe de chemin représente la succession des concepts et des relations (binaires) qui séparent deux concepts recherchés, la longueur du chemin le nombre de relations les séparant. Par exemple, lorsque l'utilisateur requête $R(c_1, c_2)$, Corese peut également construire la requête $R(c_1, c_i), R(c_i, c_2)$ si l'utilisateur a spécifié une taille de chemin de deux, ou encore $R(c_1, c_i), R(c_i, c_j), R(c_j, c_2)$ s'il l'a spécifiée à trois. En cas de résultats multiples, le comportement par défaut de Corese est de s'arrêter au premier résultat ayant le plus court chemin, mais il est tout aussi possible de tous les récupérer, dans la limite d'une asymptote combinatoire évidente – d'où la présence du nombre d'arcs à parcourir dans la requête.

Ainsi, comme on l'a vu, Corese constitue une piste solide pour approximer les requêtes utilisateurs et tenir compte, dans une certaine mesure, de la complexité d'un domaine. En effet, il permet de compenser les divergences entre les vocabulaires, avec l'approximation ontologique, et de compenser la divergence structurelle qui peut exister entre utilisateur et concepteur, avec l'approximation structurelle. Un point particulièrement intéressant est l'introduction d'une sorte d'explication pour les utilisateurs lorsque les résultats leur sont présentés ; la mesure de similarité des résultats à une requête (cf. page 61 sur la projection approximative) est utilisée pour trier les résultats du plus pertinent au moins pertinent, lorsqu'ils sont affichés à l'utilisateur.

Une autre approche pour renforcer l'approximation des requêtes réside sans doute dans la manière d'y prendre en compte l'imprécision et l'incertitude. Les travaux à propos des sous-ensembles flous et de la logique floue proposés par Zadeh (Zadeh, 1965) en 1965 constituent en cela des candidats fort pertinents.

Les ensembles flous représentent un changement de paradigme important dans la théorie des ensembles : contrairement à la théorie des ensembles classique qui définit l'appartenance des éléments à un ensemble comme une relation binaire (*i.e.* il peut, ou non, appartenir à cet ensemble), dans la théorie des ensembles flous cette appartenance perd son caractère binaire et est représentée par un degré d'appartenance à l'ensemble. Ce degré d'appartenance est défini par un intervalle de vérité, généralement continu en $[0, 1]$ où 0 signifie qu'un élément x n'appartient pas à un ensemble flou A, et 1 une appartenance **totale** : tout autre valeur comprise entre $[0, 1]$ représente la mesure dans laquelle l'élément x peut être considéré comme un élément de l'ensemble A. Le degré d'appartenance [6] d'un élément x est donné par une fonction d'appartenance (notée $\mu_A(x)$) définissant l'ensemble flou A.

La logique floue quant à elle a pour objectif de permettre de raisonner avec des événements flous régit par une valeur de vérité graduelle, et non plus binaire (Trillas et Eciolaza, 2015). Pour ce faire, elle fournit une approche par calcul compositionnel des degrés d'appartenance (Bobillo et

5. le lecteur peut se référer à (Corby et al., 2006a) pour une explication détaillée du calcul de la similarité.
6. *Membership degree*

STRACCIA, 2016), c'est-à-dire que la valeur de vérité d'une expression logique exprimée en logique floue se calcule à partir de la valeur de ses sous-expressions. Les opérations de logique classique de conjonction, de disjonction, de complément et d'implication sont ici respectivement étendues par des t-normes, de t-conormes, de négation et d'implication sur un ensemble flou. La particularité ici est les opérations appliquées sur ces ensembles peuvent être choisies en fonction des propriétés recherchées. Par exemple, la conjonction d'une règle floue $x \wedge y$ peut se calculer par $min(\mu_A(x), \mu_A(y))$ ou par $max(\mu_A(x) + \mu_A(y) - 1, 0)$, en fonction du besoin [7]. Une définition formelle de ces fonctions a été proposée par (HÁJEK, 1998).

Raisonner à l'aide de la logique floue implique plusieurs difficultés, notamment que la valeur de vérité d'une expression ne peut pas être connue précisément après son évaluation. Cela s'explique puisque l'évaluation se passe dans un intervalle continu (ou au moins non binaire) pour modéliser l'imprécision inhérente aux ensembles flous. Alors, pour passer à une décision unique, on applique une étape dite de *défuzzification* qui permet de définir la valeur de vérité issues de l'agrégation des conclusions obtenues à partir de la résolution compositionnelle des règles floues. Là aussi, il existe plusieurs méthodes, comme la moyenne des maxima ou celle du centre de gravité, chacune possédant ses caractéristiques (LEE, 1990).

En 1971, Zadeh introduit également la notion de sémantiques quantitatives floues (*quantitative fuzzy semantics*) (ZADEH, 1971). Il présente comment des termes linguistiques peuvent être exprimés sous la forme d'ensembles flous appartenant à l'univers du discours. Il présente aussi comment ces termes, ainsi que leurs combinaisons logiques, peuvent avoir des équivalents numériques et être opérés d'après les opérations théoriques associées sur ces ensembles flous (DUBOIS et PRADE, 1991). L'objectif est de pouvoir manipuler mathématiquement des termes d'un discours, comme "apprenant actif".

Dans ce courant de modélisation floue du discours, l'on voit également apparaître la notion de modificateur linguistique flou (*fuzzy modifier*), qui concerne certains termes du discours (ZADEH, 1971 ; KERRE et DE COCK, 1999). Ces modificateurs modifient les conditions d'appartenance à un terme donné. Par exemple, un étudiant avec une valeur d'activité x pourrait appartenir à l'ensemble (des étudiants) **actif** tel que $\mu_{actif}(x) > 0$, mais non à l'ensemble **très actif**, modifié par le modificateur flou m **très**, de telle sorte que $m \circ \mu_{actif}(x) = 0$ (l'on parle alors d'un modificateur restrictif ; s'il avait renforcé l'appartenance, on l'aurait dit expansif).

Comme on le devine, toutes ces propriétés peuvent être exploitées pour renforcer la manière dont les requêtes utilisateurs sont interprétées et exploitées par un système. Pour citer deux travaux qui exploitent la logique floue et qui vont en ce sens, il y a le système PASS (*Personalized Abstract Search Services*) (WIDYANTORO et YEN, 2001) et le raisonneur fuzzyDL (BOBILLO et STRACCIA, 2016). PASS est destiné à rechercher des résumés d'articles de recherche et est fondé sur un principe d'affinage (manuel) de la requête utilisateur. Ici, l'utilisateur peut remplacer un terme de sa requête par un autre avec lequel il partage une relation (*e.g.* le nouveau terme est plus précis). Pour trouver quels termes sont en relation les uns avec les autres, PASS construit une ontologie de leurs relations. Pour la construire, le système estime les relations des termes à l'aide de la logique floue, en calculant leur degré d'appartenance μ aux articles d'après une fonction de fréquence, puis en appliquant entre autres une t-norme terme à terme. De cette manière, le système définit une proximité sémantique entre les termes. Cela fait d'ailleurs écho à la manière dont la propriété *rdfs :seeAlso* est utilisée par Corese pour approximer la recherche.

fuzzyDL (BOBILLO et STRACCIA, 2016) est l'un des premiers raisonneurs à base d'ontologies floues. Il se base sur les logiques de description floues (*fuzzy DL*), initialement proposées par Yen (YEN, 1991) – rappelons ici que les raisonneurs classiques utilisent des logiques de description (DL) non floues. Pour ce faire, les propriétés dans ce type d'ontologies possèdent en plus un degré d'appartenance, représentant le degré de vérité de la relation entre deux concepts. Pour résoudre une requête utilisateur dans un tel contexte, le système a recours à la logique floue. L'avantage de ce type de système est de pouvoir explorer des propriétés et des classes qui ne correspondent pas exactement aux termes de l'utilisateur, tout en assurant une certaine pertinence lors de l'exploration dans le graphe et

7. Ces T-normes sont respectivement dites de Zadeh et de Łukasiewicz. Leur choix résulte des propriétés souhaitées, ainsi que du domaine d'application.

du raisonnement : dans un raisonneur classique bi-valué, il est nécessaire d'adopter des méthodes détournées, comme l'illustre Corese (Corby et al., 2006a).

Pour conclure, ces travaux montrent que l'approximation des requêtes constitue un enjeu majeur pour satisfaire au mieux les attentes des utilisateurs, et ainsi limiter le fossé sémantique qui peut résulter des différentes modélisations. Bien que ces techniques aient été présentées succinctement, cela permet toutefois de montrer la manière dont un tel problème peut être abordé dans le contexte de la capitalisation – où différents acteurs, ainsi que leur vision propre de l'analyse, se côtoient.

En effet, ces techniques comme la logique floue ou l'approximation ontologique offrent un socle solide pour promouvoir l'assistance aux acteurs lors de l'élaboration et l'exploitation d'analyses mises en commun. Elles vont par exemple permettre d'étendre la définition initiale de ces analyses, ou encore offrir des mécanismes de recherche plus élaborés et permettre de mieux représenter et comprendre le besoin d'analyse de ces acteurs.

Synthèse

5

Sommaire

Section	**Introduction**	**65**
Section 5.1	**Classifications des propriétés des modèles et des outils d'analyse**	**65**
Section 5.2	**Verrous scientifiques**	**72**

Introduction

L'étude de l'état de l'art nous a permis de présenter différentes propositions connexes à la fois aux processus d'analyse de traces d'apprentissage et à leur capitalisation. Comme nous l'avons vu, ces propositions prennent des formes variées et interviennent à différents moments, lors de la conception de l'analyse des traces et de sa mise en œuvre. De plus, et fait remarquable, ces propositions sont issues d'une grande variété de disciplines. Chacune adopte un point de vue spécifique sur les questions en lien avec la capitalisation et contribue à nous donner des pistes de recherche intéressantes ; ce spectre multidisciplinaire a également le mérite d'accentuer l'importance de la capitalisation.

Nous proposons tout d'abord une synthèse sous forme de tableaux des propriétés des outils d'analyse présentés durant l'état de l'art. Nous faisons ensuite émerger les verrous scientifiques qui sous-tendent cette thèse et qui, selon nous, doivent être abordés pour permettre une réelle capitalisation des processus d'analyse de traces d'apprentissage.

5.1 Classifications des propriétés des modèles et des outils d'analyse

Dans cette section, nous proposons de classer différentes propriétés des outils d'analyse qu'il nous a été donné d'étudier au cours de nos travaux. Il s'agit d'une synthèse non exhaustive des différents travaux et caractéristiques que nous avons jusque là présentés. Ceci afin d'aider le lecteur à identifier clairement l'état actuel des outils d'analyse concernant la capitalisation et ce qu'il nous semble être leur dynamique concernant cette propriété. Néanmoins, comme toute synthèse, l'on perd ici certaines de leurs particularités qu'il faut tout de même rappeler : il s'agit d'un ensemble hétérogène de travaux, qui ne partagent pas les mêmes approches (*e.g.* outils, langage informatique, modélisation) et ne couvrent pas les mêmes attentes (*e.g.* partage accru, rapidité de calcul) ; en cela, il est à notre sens impossible, tant la diversité est grande, d'observer une classification spécifique pour un type d'approche.

Il est donc nécessaire d'aborder cette synthèse en adoptant un point de vue affirmé. Pour s'en convaincre, prenons l'exemple de PMML. Pris individuellement, il s'agit d'un standard dédié à la représentation des modèles prédictifs et statistiques en *data mining* (cf. Section 3.2.1) : il est pour ainsi dire passif, puisqu'un modèle n'exécute et ne partage rien par lui-même. Néanmoins, c'est en l'intégrant dans les différents outils que les caractéristiques qu'il porte peuvent s'exprimer – avec un ressort différent selon comment ils l'implémentent. L'on peut ainsi évoquer l'interopérabilité apparente entre les outils d'analyse capables d'exporter et importer des processus décrits *via* PMML et non *via* leur modélisation *ad-hoc*.

Faut-il alors considérer l'outil d'analyse, ou la modélisation qui le sous-tend ? Par exemple, faut-il plutôt tenir compte d'UTL ou de son modèle DGU (cf. Section 3.1.2) ? Étonnament, cette question trouve

un écho important dans les verrous scientifiques que nous présentons ci-après, ainsi qu'une place centrale dans la problématique de la capitalisation. Pour réaliser cette synthèse, nous avons choisi d'inventorier les outils d'analyses. L'une des raisons est que, selon nous, ils représentent plus la "réalité terrain" actuelle des différentes disciplines. Nous supposons aussi qu'étant confrontés plus directement aux utilisateurs que les modèles, ils sont en conséquence plus susceptibles d'évoluer suivant leurs attentes et besoins. Toutefois, ce chapitre nous servira pour nous convaincre qu'il ne faut non pas opérer une séparation entre outils et modélisations, mais les considérer conjointement pour permettre la capitalisation des processus d'analyse.

Les propriétés des outils d'analyses présentées ici concernent les processus d'analyse de traces d'apprentissage et la capitalisation. Elles sont regroupées selons quatre aspects que sont la *conception technique* des processus d'analyse ; la dimension *communautaire* ; le déploiement de *l'information* ; leurs possibilités de réaliser des *recherches*. Ces aspects constituent selon nous des pans importants de la capitalisation et trouvent un écho dans notre problématique. Ces propriétés sont respectivement représentées par les tableaux 5.1, 5.2, 5.3 et 5.4.

Chaque cellule de chaque tableau est définie selon un couple $< couleur, symbole >$, qui traduit l'état d'implémentation d'une propriété (*i.e.* une ligne) pour un outil (*i.e.* la colonne) et l'incidence positive ou nulle qu'elle a sur la capitalisation. Plus précisément, nous définissons trois couples {✗, ∼, ✓}. Ces couples désignent respectivement que l'outil en question ne possède pas la propriété discutée et que par conséquent qu'elle n'a pas d'incidence sur la capitalisation, que l'outil présente des traces d'implémentation partielle de la propriété, permettant de constater un début d'incidence positive sur la capitalisation et qu'enfin, l'outil d'analyse possède la propriété discutée ce qui, potentiellement, favorise la capitalisation.

Comme nous l'avons vu au cours de l'état de l'art, l'aspect technique des processus d'analyse conditionne en grande partie leur faculté à être partagés, adaptés et réutilisés. Nous avons également montré que cette technicité est induite par le ou les outils d'analyse dans lesquels ils sont mis en œuvre. Le Tableau 5.1 nous permet d'illustrer certaines des propriétés intervenant sur le plan technique lors de la conception des processus d'analyse. D'une manière générale, l'on observe dans ce tableau que les outils d'analyse implémentent de nombreuses propriétés techniques. Cette couverture technique atteste ainsi d'une priorité accentuée quant à l'aspect computationnel des processus. Le détail des propriétés traitées est donné ci-dessous :

— **Format des traces non spécifiques** : Cette propriété concerne la manière dont un outil d'analyse représente ses traces ou ses données en interne. Nous considérons les formats de représentation de traces comme spécifiques lorsqu'ils sont propres à un outil (*e.g.* UnderTracks) et qui, *de facto*, ne sont pas issus d'une norme ou d'un standard. En outre, cette propriété est indépendante de l'existence de mécanismes permettant de passer d'un standard vers ces formats *ad-hoc*.

— **Indépendant du modèle des données** : Comme nous l'avons vu, la manière dont sont représentées les données dans les traces conditionne les informations qu'il va être possible d'extraire. Certains outils sont conçus pour interpréter des données encodées d'après des standards, comme xAPI, ou alors reposent sur un principe de senseur transformant les traces produites par une source dans leur modèle de traces. Il en résulte que ces outils sont utilisés sur des traces issues de sources hétérogènes, tout en limitant considérablement les étapes de prétraitement.

— **Utilisation de services externes** : Cela concerne la possibilité d'avoir recours à des opérations en dehors de l'outil d'analyse. Par exemple, des outils comme Taverna peuvent utiliser des webhooks pour lancer des calculs à distance et, une fois terminés, récupérer les résultats.

— **Liste les opérations disponibles** : Lister les opérations consiste à avoir dans l'outil un endroit où toutes les opérations utilisables sont accessibles. Pour un utilisateur, cela revient en quelque sorte à disposer d'un entrepôt local des opérations qu'il pourra utiliser.

— **Classifie les opérations disponibles** : Il est intéressant d'observer que certains outils d'analyse adoptent des classifications de leurs opérations. Bien que ces classifications ne soient pas homogènes entre les outils, cela revient à affecter ces opérations à des taxons et à leur attribuer une sémantique auprès des utilisateurs : les utilisateurs s'approprient alors un vocabulaire de l'analyse propre à l'outil.

— **Variables prérequises aux PA identifiées** : Lorsqu'un processus d'analyse est mis en œuvre, il l'est initialement à partir de certains types de données (*i.e.* variables). Cela revient à dire que pour pouvoir utiliser de nouveau, et correctement, tel processus d'analyse, il est nécessaire de disposer au préalable des variables adéquates. Autrement dit, que les nouvelles variables respectent les caractéristiques de celles initiales. Cette propriété illustre donc si les outils sont en mesure d'identifier ces variables nécessaires à l'exécution des processus d'analyse.

— **Imbrication native des PA** : Cette propriété mise en avant par Baker & *al.* (R. S. J. D. BAKER et YACEF, 2009) consiste à permettre la réutilisation d'un processus d'analyse déjà développé dans un autre. Cela peut se faire par exemple par l'intermédiaire du chargement d'un fichier. Néanmoins, il ne s'agit pas ici de considérer un processus d'analyse comme un opérateur, avec des paramètres, des variables d'entrée et de sortie clairement identifiées. Sinon seuls R, Jupyter et Taverna possèderaient vraiment cette propriété.

— **Importer des PA d'autres outils** : Cette propriété consiste à savoir si un outil d'analyse est capable d'importer d'autres processus d'analyse, réalisés dans d'autres outils d'analyse. Cela revient en quelque sorte à juger de l'indépendance technique des processus d'analyse développés, voire de leur interopérabilité.

— **Abstraction supplémentaire** : Affranchir les processus d'analyse d'une représentation technocentrée permet de faciliter la compréhension et d'étendre le nombre potentiel d'acteurs capable de les réutiliser. Parmi les différents types d'abstraction, l'on peut citer l'approche par workflows, ou encore par notebook.

— **Visualisation des résultats** : Cette propriété concerne la possibilité native d'explorer graphiquement les résultats générés suite à une série d'opérations.

— **Modèle sous-jacent des processus** : Dans cette propriété, nous listons le ou les modèles qui sont utilisés pour modéliser les processus d'analyse. Cela permet de faire une correspondance également avec l'état de l'art. Toutefois, lorsque les modèles utilisés ne sont pas communiqués – ce qui est principalement le cas pour les solutions propriétaires – ou n'existent pas, comme dans R, nous l'avons indiqué par une barre oblique : '/'.

Aspect : **Conception Technique**	Orange : D.M.	UnderTracks	UTL	RapidMiner	Tigris	kTBS	SPSS	R	Jupyter	Galaxy	Taverna
Format des traces non spécifique	✗	✗	✗	✓	✓	✗	✓	✓	✓	✓	✓
Indépendant du modèle des données	✗	✗	✓	✗	∼	∼	✗	✓	∼	✗	✗
Utilisation de services externes	✗	✗	✗	✓	✗	✓	✗	✓	✓	✓	✓
Liste les opérations disponibles	✓	✓	✗	✓	✓	✗	✓	✗	✗	✓	✓
Classifie les opérations disponibles	✓	✓	✗	✓	✓	✗	∼	✗	✗	✓	✓
Variables prérequises aux PA identifiées	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✓
Imbrication native des PA	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓
Importer des PA d'autres outils	✗	✓	✗	✗	∼	✗	✗	✓	✓	✗	✗
Abstraction supplémentaire	✓	✓	✗	✓	✓	✗	✗	✗	✓	✓	✓
Visualisation des résultats	✓	✓	∼	✓	✓	✗	✓	✓	✓	✓	✓
Modèle sous-jacent des processus	/	DOP8	DGU	/	/	kTBS	/	/	iPython	/	wf4ever

Tableau 5.1.: Classification de propriétés techniques pouvant être associées à la capitalisation des processus, au sein de différents outils.

Aspect : **Communautaire**	Orange : D.M.	UnderTracks	UTL	RapidmMiner	Tigris	kTBS	SPSS	R	Jupyter	Galaxy	Taverna
Intégré à un entrepôt communautaire de traces	✗	✓	✗	✗	✓	∼	✗	✗	✗	✓	✗
Intégré à un entrepôt communautaire de PA	✗	✓	✗	∼	✓	✗	✗	✓	✗	✓	✓
Distingue différents acteurs (créateur, utilisateurs...)	✗	✗	✓	✗	✗	✗	✗	✗	✗	∼	✓
Système de reviewing	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓
Système de versionning	✗	✗	✓	✗	✗	✗	✗	✗	✗	∼	✓
Mise en relation avec d'autres PA	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗

Tableau 5.2.: Classification de propriétés communautaires intégrées à différents outils d'analyse pouvant être associées à la capitalisation des processus, au sein de différents outils.

L'intégration d'une composante communautaire à certains outils d'analyse a, comme nous l'avons vu, permis d'étendre la visibilité des analyses et d'améliorer leur possibilité d'être partagées. Le Tableau 5.2 synthétise certaines des propriétés saillantes de cet aspect communautaire pouvant contribuer à renforcer la capitalisation des processus d'analyse. Contrairement à la conception des processus où une certaine consistance des propriétés entre les outils pouvait être observée, l'on remarque une forte disparité concernant l'aspect communautaire des outils. Le détail des propriétés traitées est donné ci-dessous :

— **Intégré à un entrepôt communautaire de traces** : Cette propriété indique s'il est possible pour un outil d'accéder à un entrepôt stockant des traces. La particularité ici de cet entrepôt est de permettre à différents acteurs de la communauté de pouvoir les y déposer. Un exemple est DataShop, qui est nativement intégré à Tigris.

— **Intégré à un entrepôt communautaire de PA** : Au même titre que celle concernant les traces, cette propriété indique si un outil peut accéder et réutiliser des processus d'analyse développés par la communauté. L'objectif est d'identifier les outils qui tentent de favoriser la collaboration communautaire. Les "marketplaces" propriétaires, gérés uniquement par les éditeurs des outils, qui ne laissent que peu de place à la collaboration, sont indiqués par ∼

— **Distingue différents acteurs** : Dans une approche communautaire, il devient intéressant d'identifier clairement les différents acteurs impliqués dans l'analyse, voire le rôle qu'ils y ont joué, tout cela avec une granularité plus ou moins marquée. Outre apporter une certaine crédibilité scientifique, cela contribue à renforcer l'interaction communautaire. L'on voit ainsi apparaître des distinctions entre créateur du processus et utilisateurs, ou encore entre créateur et examinateur du processus d'analyse.

— **Système d'évaluation** : Cette propriété représente s'il est possible d'évaluer, d'une quelconque manière, un processus d'analyse qui a été partagé. Cette évaluation peut se faire de plusieurs manières, comme à l'aide d'un système de score, ou bien avec des commentaires.

— **Système de versionnage** : Cette propriété traite de la possibilité pour un outil de sauvegarder les différetes modifications d'un même processus d'analyse commes des versions différentes. Concernant Galaxy, le versionning des processus est actuellement un travail en cours (Beek, 2018).

— **Mise en relation avec d'autres PA** : Cette propriété consiste à mettre en lien les processus d'analyse avec d'autres, connexes ; par exemple, lorsqu'un processus est similaire à un autre. L'intérêt d'une telle propriété est de renforcer la visibilité des analyses, et la navigabilité, pour par exemple renforcer la sérendipité.

Aspect : **Informations**	Orange : D.M.	UnderTracks	UTL	RapidmMiner	Tigris	kTBS	SPSS	R	Jupyter	Galaxy	Taverna
Champ de description du PA	✗	✓	✓	✗	✗	✗	✗	✗	✓	✓	✓
Commentaires	✓	✓	✗	✓	✓	✗	✓	✓	✓	✓	✗
métadonnées de publications	✗	✓	✓	✓	✓	✗	✗	∼	✓	✓	✓
métadonnées étendues	✗	✗	✗	✗	✗	✗	✗	✗	∼	✗	∼
Intégration de termes sémantiques	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓	✓
Relations entre les éléments du PA	✗	✗	∼	✗	✗	✓	✗	✗	✗	✗	∼
Éléments décrivant le contexte du PA	✗	✓	✗	✗	✗	✗	✗	✗	∼	∼	✗
Éléments/objets supplémentaires	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✓

Tableau 5.3.: Tableau classifiant différentes propriétés descriptives des outils d'analyse pouvant être associées à la capitalisation des processus d'analyse de traces.

Aspect : **Recherche & Requêtes**	Orange : D.M.	UnderTracks	UTL	RapidmMiner	Tigris	kTBS	SPSS	R	Jupyter	Galaxy	Taverna
Recherche native dans les traces	✗	✓	✗	✗	∼	∼	✓	∼	✗	✓	✓
Recherche native dans les PA	✓	✓	✗	✓	✓	✗	✗	✗	✗	✓	✓
Filtres de recherche	✗	✗	✗	✗	✓	✓	✓	∼	✗	✓	✓
Recherche avancée	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Approximation sémantique	✗	✗	✗	✗	✗	∼	✗	✗	✗	✗	∼
Possibilité d'utiliser des termes du vocabulaire	✗	✗	✗	✗	✗	∼	✗	✗	✗	✗	∼

Tableau 5.4.: Tableau classifiant différentes facultés de recherche intégrées à différents outils d'analyse.

Afin de renforcer la compréhension des analyses et la manière dont elles sont mises en œuvre, nous avons vu que différents outils permettent de leur ajouter des informations. Le spectre des informations qu'il est possible d'y décrire, ainsi que le type de ces dernières, est néanmoins vaste, et non uniformisé. Toutefois, en adoptant une vision plus généraliste, il est possible de catégoriser certaines des propriétés descriptives de ces outils, comme l'illustre le Tableau 5.3. L'on constate ainsi que la plupart des outils fournissent des moyens descriptifs de base, et qu'une certaine corrélation existe entre l'aspect communautaire et descriptif des outils : cela atteste selon nous de l'importance de la description comme vecteur de compréhension des analyses et de partage par la communauté. Le détail des propriétés traitées est donné ci-dessous :

— **Champ de description du PA** : Cette propriété représente s'il est possible de décrire le processus d'analyse dans son ensemble, par l'entremise d'un champ dédié de description. Nous ne tenons ici pas compte de la granularité de la description dans ce champ, ni de la manière dont cette description est réalisée (*e.g.* texte libre), mais juste de la possibilité qu'a un acteur d'apporter des informations supplémentaires sur l'analyse (*e.g.* son besoin).

— **Commentaires** : Les commentaires sont des annotations – généralement courtes – utilisées pour expliquer des parties du processus, des opérations précises ou encore des choix lors de l'implémentation.

— **Métadonnées de publication** : Cette propriété concerne la présence d'informations en lien direct avec la publication d'un processus d'analyse. Cela concerne par exemple l'auteur du processus d'analyse, ou sa date de publication. L'on peut également citer la licence sous laquelle le processus est publié. Les outils permettant de renseigner au moins deux métadonnées de publications ont été classés positifs.

— **Métadonnées étendues** : Il s'agit de la présence d'informations supplémentaires et plus élaborées que simplement les métadonnées de publication, et qui sont en lien avec le processus d'analyse ou ses éléments constitutifs. Par exemple, nous considérons les hypothèses scientifiques comme des métadonnées étendues. Les outils permettant de renseigner au moins deux types de métadonnées étendues sont classés positifs ; ceux permettant d'en renseigner d'un seul type sont indiqués par ∼. Notez que nous avons ici séparé métadonnées étendues et objets associés aux processus (*e.g. Research Object*), ce qui explique pourquoi Taverna/^{my}Experiment n'est pas classé positif.

— **Intégration de termes sémantiques** : Cette propriété indique si l'outil en question utilise des termes sémantiques déréférencables. Nous n'avons pas opéré de distinction entre traces et processus d'analyse, ceci pour permettre d'identifier la dynamique générale en lien avec cette propriété.

— **Relations entre les éléments du PA** : Cette propriété représente si un outil d'analyse est capable de définir des relations sémantiques entre les différents éléments constitutifs du processus d'analyse.

— **Éléments décrivant le contexte du PA** : Cette propriété concerne la présence d'un champ dédié à la description du contexte de l'analyse et de sa mise en œuvre. Autrement dit, avoir la possibilité de décrire pour un processus d'analyse quels acteurs sont concernés, les différentes situations pédagogiques, *etc*.

— **Éléments/objets supplémentaires** : Cela concerne la possibilité de pouvoir joindre aux processus des artefacts supplémentaires, comme des graphiques résultats ou des échantillons des données utilisées. C'est par exemple le cas dans Taverna/^{my}Experiment grâce aux *Research Object*. En revanche, ni la qualité de la relation, ni le type d'artefact n'est ici pris en compte.

Assister les utilisateurs dans leur tâche en lien avec l'analyse est une opération complexe. Ce que l'on remarque est que la principale source d'assistance fournie aux utilisateurs réside dans la recherche des analyses. Aussi, le Tableau 5.4 propose de comparer les possibilités de recherche que fournissent les outils aux utilisateurs. L'observation de ce tableau ne manquera pas de faire remarquer au lecteur l'importante scission entre des mécanismes de recherche que nous qualifions de basiques (les trois premières propriétés) et ceux faisant intervenir des mécanismes plus complexes, mais *a fortiori* plus puissants, comme nous l'avons vu. Le détail des propriétés traitées est donné ci-dessous :

— **Recherche basique dans les traces** : Cette propriété concerne la possibilité de pouvoir rechercher une information quelconque dans les traces. Nous classifions comme basique le fait de rechercher uniquement une correspondance textuelle entre la requête et les objets consultés (*e.g.* appliquer une expression régulière). Nous ne prenons pas en compte ici la possibilité d'utiliser des étapes de prétraitement supplémentaires pour explorer les traces ; nous nous intéressons uniquement aux fonctionnalités internes à l'outil de parcourir les traces chargées.

— **Recherche basique dans les PA** : Similairement à la propriété précédente, il s'agit d'identifier si un outil peut rechercher une information dans les opérations qu'il possède, que ce soit localement ou dans un entrepôt. Généralement, la comparaison s'effectue avec le nom de l'opération ; nous ne considérons pas que rechercher dans des métadonnées ou directement dans la structure des processus d'analyse relève d'une propriété de recherche basique, puisqu'un travail de conception interne sur la modélisation est nécessaire.

— **Filtres de recherche** : Cela concerne la possibilité d'utiliser des modificateurs de recherche pour enrichir, spécialiser ou généraliser la recherche, et modifier les résultats qu'elle produit. Par rapport à une recherche basique, des options supplémentaires de recherche sont alors disponibles, conjoitement à la requête utilisateur. Pour ce faire, le système peut s'appuyer sur des métadonnées. Par exemple, un utilisateur pourrait ne conserver que des processus d'analyse qui ont été évalués positivement.

— **Recherche avancée** : Dans la majorité des outils, la recherche s'effectue en considérant des informations basiques comme le nom ou encore les auteurs, comme nous l'avons vu dans l'état de l'art. Ici, cette propriété consiste à identifier les outils qui permettent d'explorer des éléments supplémentaires lors de la recherche, comme une configuration spécifique d'un processus d'analyse, ou encore des artefacts supplémentaires comme la présence de graphiques.

— **Approximation sémantique** : Cette propriété concerne la possibilité qu'a un outil à utiliser un réseau sémantique pour trouver des solutions lors de la recherche. Cela permet à l'outil d'obtenir des résultats qui ne sont pas exactement identiques aux termes utilisés dans la requête utilisateur, notamment *via* l'utilisation de règles de subsomptions, ou encore de transitivité.

— **Possibilité d'utiliser des termes du vocabulaire** : Cette propriété concerne les outils qui permettent à un utilisateur d'utiliser des termes spécifiques lorsqu'il décrit sa requête, contenus par exemple dans un dictionnaire qui est mis à sa disposition.

5.2 Verrous scientifiques

En reconsidérant l'état de l'art, nous pouvons identifier un grand nombre de difficultés sous-jacentes à la capitalisation des processus d'analyse de traces d'apprentissage, qui ne sont pas encore résolues. Ainsi, la diversité pédagogique et la diversité des traces est encore mal prise en compte dans l'analyse (cf. Tableau 5.1), tout comme leurs impacts. La présence de standards de traces d'e-learning n'est d'ailleurs pas exploitée pour renforcer cet aspect. Et même, de manière générale, le contexte de l'analyse est absent des processus d'analyse, alors qu'il s'agit pourtant d'une contrainte importante pour la capitalisation (cf. Section 3.1.3, page 31). Cela s'explique majoritairement par le fait que les outils d'analyse existant actuellement ont des impératifs techniques très forts, qui induisent des caractéristiques affectant directement les processus d'analyse et la manière dont ils peuvent être mis en œuvre (cf. Section 2.2, page 18). La dépendance des outils (cf. Tableau 5.1) vis-à-vis du format de leurs données en est un exemple – c'est d'ailleurs ce qui peut expliquer pourquoi la plupart des outils présentés ne sont pas intégrés à un entrepôt communautaire de traces (cf. Tableau 5.2).

La dépendance technique qui en résulte et qui empêche de réutiliser les processus d'analyse dans d'autres contextes techniques et pédagogiques s'observe particulièrement bien dans les outils communautaires (cf. Chapitre 3). Bien que mettant à disposition de toute une communauté les analyses que ses membres peuvent mettre en œuvre, celles-ci ne sont que rarement réexploitées. Puisque la spécificité des outils utilisés s'exprime, il devient en effet difficile d'adaptater les processus d'analyse pour d'autres situations.

Pour d'abord combattre cette dépendance technique, de nombreux travaux ont proposé de nouveaux modèles (*e.g.* PMML, cf. Section 3.2.1, page 34) pour représenter les processus d'analyse. Bien que

constituant une avancée majeure, ces modèles sont néanmoins conçus pour répondre à ce prérequis computationnel. Paradoxalement, ils reposent donc encore sur les capacités des outils d'analyse pour s'exprimer ; pour certains, leur modélisation ne couvre pas non plus l'ensemble du spectre des analyses. Dès lors, comme l'illustre le Tableau 5.3, les informations qu'il est possible d'intégrer aux processus d'analyse sont toujours grandement limitées, ainsi que leur complexité.

Comme nous l'avons vu, certaines modélisations sémantiques apportent quelques éléments de réponse concrets pour s'affranchir un peu plus de la dépendance technique (cf. Section 4.1, page 42). Toutefois, des lacunes sont notables, comme une séparation entre données et processus trop hermétique, alors qu'ils véhiculent tous deux des informations contextuelles non négligeables ; comme le fait que ces modélisations ne contribuent pas non plus à renforcer les possibilités d'adaptation des processus. De telles modélisations considèrent que pouvoir adapter un processus s'établit comme une propriété naturelle, dès lors qu'un nouveau besoin se manifeste ; or, il nous semble évident qu'il s'agit d'une équation faisant intervenir d'une part le nouveau besoin – et son contexte pédagogique – et d'autre part l'expertise des acteurs impliqués, le processus d'analyse à réutiliser, sa qualité d'implémentation et les informations qui y ont été intégrées. Cela s'observe sur la faculté de mettre les processus en relation entre eux (cf. Tableau 5.2) et avec d'autres éléments (cf. les trois dernières propriétés du Tableau 5.3).

Enfin, malgré la diversité des modélisations qui peuvent concerner cette capitalisation, l'on remarque étonnament une absence de mécanismes d'assistance pour l'élaboration et l'exploitation des processus d'analyse à destination des différents acteurs – qui sont eux-mêmes absents des modélisations. Le Tableau 5.4 permet de s'en rendre compte, notamment avec les trois dernières propriétés : les mécanismes de recherche ne sont que superficiels, n'exploitent pas la richesse des processus d'analyse et ne prennent pas non plus en compte l'expertise des acteurs (cf. troisième propriété du Tableau 5.2). Toutefois, comme le laissent transparaître les travaux vus en Section 4.2, il est possible d'exploiter la sémantique pour assister les utilisateurs dans cette démarche de capitalisation.

Pour dépasser ces limites, et ainsi répondre à la problématique posée (cf. page 2), il est nécessaire de s'intéresser aux trois verrous scientifiques suivants :

Verrou scientifique 1

Comment rendre les processus d'analyse de traces d'apprentissage techniquement indépendants des outils d'analyse, tout en homogénéisant leur modélisation pour tendre vers un formalisme fédérateur ?

Verrou scientifique 2

Comment formaliser les processus d'analyse de traces d'apprentissage pour les rendre intrinsèquement compréhensibles, adaptables et réutilisables par l'Humain ?

Verrou scientifique 3

Comment assister les différents acteurs de l'analyse en exploitant la sémantique véhiculée dans les processus d'analyse de traces d'apprentissage et les informations qui y sont associées ?

Enfin, nous avons aussi constaté que la capitalisation des processus d'analyse ne possède pas de définition claire. Elle est amalgamée avec des notions telles que la reproductibilité et la réutilisabilité. De plus, nous remarquons une absence de considération de la pluridisciplinarité des acteurs impliqués dans l'analyse et de leur expertise, comme l'atteste le Tableau 5.2. Dès lors, et avant de pouvoir s'intéresser aux trois verrous susmentionnés, il convient de se préoccuper du verrou suivant :

Verrou scientifique 4

Comment caractériser la capitalisation, ainsi que les rôles, les interventions et l'impact des différents acteurs intervenant lors de l'analyse de traces d'apprentissage ?

> *De tout ceci il résulte que les pensées déposées sur le papier ne sont rien de plus que la trace d'un piéton sur le sable. On voit bien la route qu'il a prise ; mais pour savoir ce qu'il a vu sur la route, on doit se servir de ses propres yeux.*
>
> — **Arthur Schopenhauer**
> (Extraits des *Parerga et Paralipomena*)

Partie II

Contributions théoriques

Introduction & Plan de la partie

Dans cette partie, nous présentons nos contributions théoriques qui s'illustrent par différents modèles, méta-modèles, ainsi qu'à travers des règles d'inférence et des algorithmes. Ces propositions contribuent à répondre aux difficultés observées tout au long de la Partie I concernant la capitalisation des processus d'analyse de traces d'apprentissage dans le domaine des Learning Analytics.

En effet, comme nous l'avons observé *via* l'état de l'art, les processus d'analyse sont dépendants du contexte technique dans lequel l'analyse a été mise en œuvre. Cela comprend, entre autres, l'outil d'analyse utilisé, les configurations disponibles pour les opérations, ou encore le format des traces analysées. Cette dépendance se matérialise par des spécificités fortes, qui ne sont pas et ne peuvent pas être partagées par les autres outils d'analyse. Dès lors, ne serait-ce qu'une propriété de partage des processus d'analyse est difficilement envisageable. Or, les différents modèles de processus d'analyse proposés dans la littérature et qui s'inscrivent dans cette mouvance d'un paradigme entièrement dédié à répondre à un besoin computationnel ne semblent pas permettre de s'affranchir de cette dépendance technique.

En outre, nous avons également pu observer la complexité du contexte pédagogique et l'impact qu'il peut avoir lors de l'analyse. Mais nous avons aussi vu que ce contexte est actuellement difficile à modéliser dans les processus d'analyse de traces, notamment parce que peu d'informations y sont représentables - et encore moins de manière structurée : certaines informations importantes sont implicites dans l'analyse ; d'autres, absentes. Cela provoque des ambiguïtés de compréhension, laissant ainsi la place à des approximations et des incertitudes quant à la réutilisation des processus pour répondre à de nouveaux besoins d'analyse - dans les rares cas où ils sont mis entièrement à disposition.

Ces deux types de contraintes réunis exigent des différents acteurs de l'analyse, pour pouvoir réutiliser un processus d'analyse, une expertise poussée de l'analyse, certes, mais aussi une expertise des outils d'analyse disponibles, des différents formats de traces, des différentes modélisations de processus d'analyse (*e.g.* workflows) et des différents contextes pédagogiques. Exigence qui, selon nous, est utopique par son étendue. De plus, il faut rappeler que les différents acteurs qui interviennent dans l'élaboration de l'analyse (*e.g.* enseignants, scientifiques) possèdent des expertises variées qui ne concernent pas toujours l'analyse. Par conséquent, les acteurs non experts de l'analyse peuvent difficilement réexploiter l'existant, surtout que les outils d'analyse actuels ne permettent pas de les assister lors de l'élaboration, du partage, de l'adaptation ou de la réutilisation des processus. Pour cela, il est nécessaire de changer le paradigme actuel des processus d'analyse des traces d'apprentissage.

Dans l'optique de construire un paradigme des processus d'analyse permettant naturellement leur capitalisation, notre travail se veut itératif. Dans un premier temps, il convient de montrer qu'il est possible d'émanciper les processus d'analyse des dépendances techniques générées par les outils et les traces. Cet effort réside dans notre approche **CAPTEN-MANTA** (cf. nomenclature Figure 1.1 page 5). Dans cette partie, nous en présentons, dans le Chapitre 7, sa partie théorique, à savoir **CAPTEN-ALLELE**, qui regroupe l'ensemble des méta-modèles qui permettent cette émancipation.

Dans un deuxième temps, et tout en renforçant cette indépendance technique, il convient de montrer qu'intégrer et structurer directement l'information dans les processus d'analyse en suivant une démarche narrative permet la capitalisation. Cette proposition est portée dans notre approche par

CAPTEN-ATOM. Nous en présentons, dans le Chapitre 8, la partie théorique de cette approche, à savoir **CAPTEN-ONION**, qui introduit principalement une ontologie pour représenter les analyses.

Enfin, nous montrons qu'il est possible d'exploiter cette information pour fournir des assistances innovantes à destination des différents acteurs lors de l'élaboration et de la réutilisation des processus d'analyse. Dans le Chapitre 9, nous parlons de **CAPTEN-AERIS**, l'assistance à la recherche dans les processus d'analyse. Nous y présentons les mécanismes et les règles d'inférence utilisés pour mener à bien cette assistance, regroupés sous la dénomination **CAPTEN-FRUIT**. D'autres propositions d'assistances seront évoquées dans les perspectives, ce qui nous permet d'envisager un ensemble d'assistance pour la capitalisation.

Avant cela, il faut aussi noter que les termes de capitalisation, de reproductibilité et de réutilisation sont polysémiques. Conscients de cette ambiguïté, nous présentons dans le Chapitre 6 notre définition de la capitalisation et de ses propriétés. Nous y proposons également un cycle d'élaboration et d'exploitation des processus d'analyse par les différents acteurs. L'intention ici est de formaliser les différents acteurs, leurs rôles et les relations qu'ils partagent, pour comprendre comment tout cela s'ancre au sein de la capitalisation des processus d'analyse.

Concepts fondamentaux de notre approche

6

Sommaire

Section	**Introduction**	**79**
Section 6.1	**Un cycle d'élaboration et d'exploitation des processus d'analyse issu de la littérature**	**79**
6.1.1	Les différents acteurs et étapes de l'analyse	80
6.1.2	Illustration	83
6.1.3	Difficultés à réutiliser des analyses	84
Section 6.2	**La capitalisation et ses propriétés intrinsèques**	**86**
Section 6.3	**Notre approche pour capitaliser**	**88**

Publications relatives à ce chapitre

(Lebis et al., 2018a) A. Lebis, M. Lefevre, V. Luengo et N. Guin (2018a). "Capitalisation of Analysis Processes : Enabling Reproducibility, Openess and Adaptability thanks to Narration". In : *LAK '18 - 8th International Conference on Learning Analytics and Knowledge*. Sydney, Australia : ACM, p. 245–254

(Lebis et al., 2017a) A. Lebis, M. Lefevre, V. Luengo et N. Guin (2017a). "Approche narrative des processus d'analyses de traces d'apprentissage : un framework ontologique pour la capitalisation". In : *Environnements Informatiques pour l'Apprentissage Humain*. EIAH 2017. Strasbourg, France

Introduction

Dans ce chapitre, nous présentons les concepts fondamentaux qui constituent notre approche pour permettre la capitalisation des processus d'analyse de traces d'apprentissage. Afin d'être en mesure de la présenter convenablement, nous proposons tout d'abord en Section 6.1 un cycle d'élaboration et d'exploitation des analyses de traces d'apprentissage issu de l'étude de la littérature. Ensuite nous mettons en avant dans la Section 6.2 la présence d'une polysémie sur les termes de reproductibilité, de réutilisabilité et de réplicabilité. Nous en proposons des définitions et des formalismes, et les intégrons au sein de la capitalisation, que nous définissons également comme un ensemble de propriétés interdépendantes. Enfin, la Section 6.3 introduit notre approche, et nous permet de faire pressentir son impact dans le cycle actuel d'élaboration et d'exploitation des analyses.

6.1 Un cycle d'élaboration et d'exploitation des processus d'analyse issu de la littérature

Nous l'avons vu, la capitalisation n'est réellement effective que lorsque les ressources qui sont partagées peuvent être réutilisées correctement, et sans ambiguïté, en fonction des différents acteurs. Répondre à

cette nécessité pour les processus d'analyse de traces d'apprentissage requiert l'identification préalable des particularités des acteurs impliqués, ainsi que des différentes étapes de l'analyse. Dans cette optique, nous présentons un cycle décrivant l'élaboration et l'exploitation des processus d'analyse, et nous l'illustrons sur un exemple afin de montrer ensuite les difficultés pouvant survenir pour la réutilisation de processus d'analyse existants.

6.1.1 Les différents acteurs et étapes de l'analyse

L'étude de la littérature (Tsantis et Castellani, 2001 ; J. C. Stamper et al., 2011 ; Greller et Drachsler, 2012 ; Chatti et al., 2012 ; Volle, 2001 ; Bouhineau et al., 2013a) permet de se rendre compte que l'élaboration et l'exploitation d'une analyse ne sont pas issues d'une démarche monolithique, mais consiste en un ensemble d'étapes qui fait intervenir différents acteurs. Ces différentes étapes sont en relation les unes avec les autres et partagent un objectif commun, qui est de répondre à un besoin. Dans un objectif de réutilisation d'un processus d'analyse, la compréhension des liens entre ces différentes étapes de l'élaboration d'une analyse devient incontournable, et le partage des informations liées à chacune de ces étapes est nécessaire.

Nous proposons de formaliser ces étapes issues de la littérature et leurs interdépendances sous la forme d'un cycle d'élaboration et d'exploitation de l'analyse, afin de bénéficier d'une vision d'ensemble nous permettant de mieux comprendre les limites actuelles à la réutilisation des processus d'analyse de trace. Ce cycle (cf. Figure 6.1) est composé de six étapes que nous décrivons ci-dessous, à savoir :

- **E1** : l'identification et la formulation du besoin ;
- **E2** : la sélection des traces pertinentes pour l'analyse en prenant en compte le contexte d'apprentissage ;
- **E3** : la préparation de l'analyse ;
- **E4** : l'implémentation de l'analyse ;
- **E5** : la mise à disposition des résultats de l'analyse ;
- **E6** : l'utilisation des connaissances produites qui peut donner lieu à une intervention.

Nous distinguons en outre quatre rôles pour les acteurs prenant part à ces étapes :

- **R1** : le décideur, qui exprime un besoin d'analyse ;
- **R2** : l'expert de l'environnement d'apprentissage, qui connaît les traces à analyser ;
- **R3** : l'analyste, qui élabore le processus d'analyse et l'implémente ;
- **R4** : les bénéficiaires, qui utilisent les connaissances issues de l'analyse.

Notons qu'un même acteur peut être amené à endosser plusieurs de ces rôles lors de l'élaboration et de l'exploitation d'une analyse. Nous détaillons ci-dessous comment ces différentes étapes s'articulent et comment les rôles y contribuent.

Une analyse ne peut être considérée indépendamment du besoin auquel elle répond. De ce fait, il est important de prendre en compte une étape d'**identification et de formulation du besoin** (**E1**). Cette étape fait intervenir ce que nous appelons un décideur (**R1**), rôle endossé par exemple par un enseignant, une institution ou bien encore un administrateur de solution d'e-learning. L'objectif de ce décideur est de favoriser l'apprentissage et l'efficacité pédagogique des moyens mis en œuvre (*e.g.* environnement d'apprentissage) à destination d'individus en formation, le plus souvent des apprenants. Dans cette étape **E1**, le décideur décrit le besoin auquel l'analyse devra répondre – besoin potentiellement énoncé par un bénéficiaire (**R4**). Pour formuler ce besoin, le décideur identifie les informations nécessaires, comme le contexte d'apprentissage, ou les politiques de confidentialité et d'éthique. Cependant, un décideur peut ne pas être un expert en analyse et donc peut ne pas avoir conscience de ce qui est réalisable ou non. C'est l'analyste (**R3**) qui peut dire s'il est possible de répondre au besoin formulé par le décideur. Cette interaction entre le décideur et l'analyste peut amener le décideur à faire évoluer la description du besoin.

Pour terminer, cette étape de formulation du besoin est le moment où l'aspect éthique de l'analyse doit être abordé, puisque la manière de traiter le besoin va indéniablement répercuter les contraintes

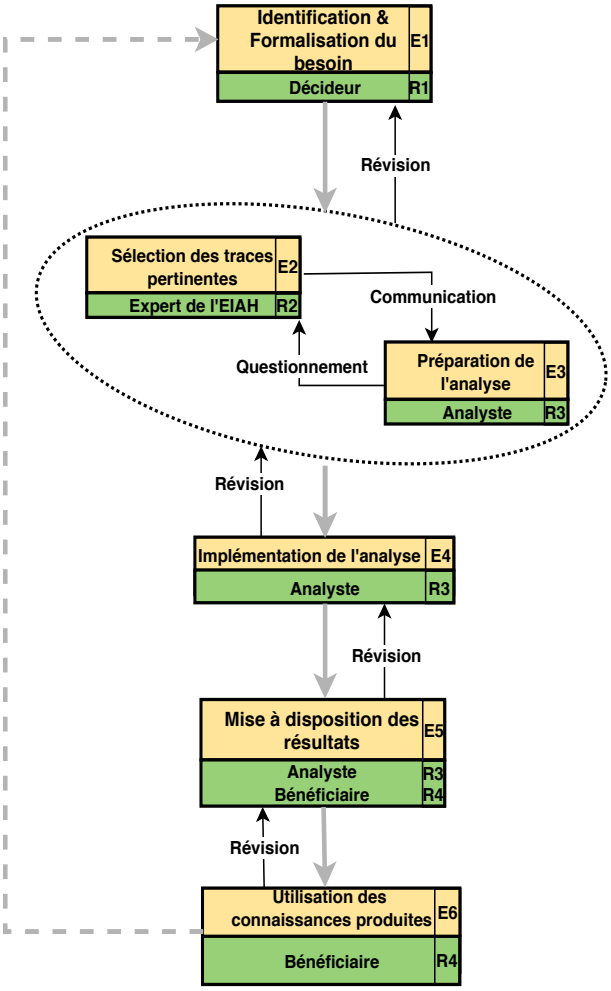


Figure 6.1.: Cycle d'élaboration et d'exploitation d'un processus d'analyse.

éthiques aux étapes suivantes. Cela peut se faire avec une consultation du bénéficiaire de l'analyse, voire de l'expert (**R2**), mais également avec les sujets sur lesquels porte l'analyse (*e.g.* apprenants) puisqu'ils sont généralement les premiers concernés. Par exemple, l'anonymisation des apprenants va nécessiter de modifier la manière dont les traces sont exportées depuis l'environnement d'apprentissage. Dans la perspective d'une capitalisation des processus d'analyse, cette dimension éthique permettrait aux sujets de l'analyse de consulter les résultats issus de chaque étape du cycle que nous présentons, par exemple quelles données leur appartenant sont utilisées dans l'analyse, ou bien encore la manière dont sont utilisées les connaissances produites pour modifier leur situation pédagogique.

Bien qu'elle soit souvent implicite dans la littérature, la seconde étape dans le cycle d'élaboration d'une analyse concerne la **sélection des traces pertinentes** (**E2**) qui serviront à répondre au besoin (Bouhineau et al., 2013a). Au sein du projet ANR HUBBLE (Projet HUBBLE, 2016), nous avons constaté à plusieurs reprises qu'un échange entre le décideur (**R1**) et l'analyste (**R3**) n'étaient pas suffisants pour mener à bien une analyse. En effet, le décideur ne possède pas toujours l'expertise nécessaire sur l'environnement d'apprentissage qui lui permettrait d'en extraire des jeux de traces adéquates pour répondre au besoin. Il est par exemple difficilement concevable d'attendre d'un enseignant qu'il soit capable d'extraire des informations d'une base de données d'un MOOC.

C'est pourquoi au sein du projet HUBBLE, la sélection des traces a été réalisée par celui que nous nommons l'expert de l'environnement d'apprentissage (**R2**). Ce rôle requiert de connaître l'environnement d'apprentissage dans son ensemble, afin de choisir méticuleusement quelles traces doivent être analysées, en prenant en compte la structuration pédagogique de l'EIAH. Cette étape de sélection

requiert de la part de l'expert de décrire les traces disponibles, d'identifier celles qui sont pertinentes par rapport au besoin énoncé, et de préciser la nature des variables présentes dans ces traces. Les traces ainsi sélectionnées décrivent l'activité des sujets sur lesquels porte l'analyse, qui sont généralement les apprenants utilisateurs de l'EIAH, mais aussi des traces relatant le fonctionnement de l'EIAH utilisé par ces sujets.

La troisième étape est la **préparation de l'analyse** (**E3**) – qui n'inclut pas son implémentation. Cette étape est réalisée par l'analyste (**R3**), qui peut être par exemple un statisticien ou un chercheur. L'objectif de l'analyste dans cette étape est de concevoir à un niveau conceptuel l'analyse qui servira à répondre au besoin, énoncé par le décideur et enrichi par l'expert de l'EIAH. Pour ce faire, l'analyste doit interagir avec l'expert (**R2**) pour comprendre le contexte d'apprentissage dans lequel s'inscrit le besoin exprimé par le décideur (**R1**), et pour comprendre les traces disponibles.

Il s'agit d'une étape délicate où l'analyste doit s'approprier le domaine d'apprentissage en échangeant avec l'expert de l'environnement d'apprentissage (**R2**), afin de choisir quel type d'analyse appliquer (*i.e.* descriptive, diagnostique, prédictive ou prescriptive (GARTNER, 2018)), quelle stratégie d'analyse mettre en œuvre (*i.e.* les différentes parties de l'analyse), et sur quels éléments des traces.

Cette étape de préparation permet un échange entre l'analyste et le décideur, qui peut amener à reformuler le besoin, par exemple en précisant des éléments qui n'étaient pas suffisamment spécifiés, mais aussi en décrivant de nouvelles attentes. En effet, le décideur n'est pas toujours conscient des connaissances qu'il est possible de découvrir à partir des traces, et nous avons constaté au sein du projet HUBBLE que la discussion entre le décideur, l'expert et l'analyste amène parfois le décideur à préciser le besoin ou à formuler de nouveaux besoins.

La quatrième étape consiste à **implémenter l'analyse** (**E4**) conçue dans l'étape précédente. C'est dans cette étape que l'analyste (**R3**) va analyser les traces fournies par l'expert (**R2**), pour en extraire des connaissances répondant au besoin exprimé par le décideur (**R1**). Ces connaissances peuvent être des indicateurs (*e.g.* le taux d'abandon dans un MOOC) ou des modèles (*e.g.* un modèle de l'engagement de l'apprenant).

Comme présenté dans l'état de l'art (cf. Section 2.1, page 11), la mise en œuvre de l'analyse est souvent décrite comme une succession de trois phases : le prétraitement des traces, l'application d'opérateurs d'analyse, et le post-traitement. Dans la phase de prétraitement, l'analyste prépare les traces pour qu'elles soient exploitables par les outils d'analyse qu'il compte utiliser. Cette phase de prétraitement découle directement de l'étape de préparation de l'analyse (**E3**), et peut faire intervenir l'expert (**R2**), qui indique par exemple quels attributs des traces peuvent être mis à disposition (ROMERO et al., 2010a). La phase de post-traitement permet de rendre l'information produite plus compréhensible pour les bénéficiaires (**R4**) de l'analyse.

Cette étape d'implémentation (**E4**) peut amener à revenir sur les deux étapes précédentes (*i.e.* **E2** et **E3**), puisque à la fois l'export des traces et la conception de l'analyse peuvent nécessiter d'être affinés. De plus, dans certains cas, il peut également être nécessaire de revenir à l'étape de formulation du besoin (**E1**). Il est à noter que la majorité des efforts dans le domaine des Learning Analytics concerne cette étape d'implémentation de l'analyse, tout en omettant l'interdépendance entre les étapes, ainsi qu'entre les différents rôles, contribuant de ce fait à limiter les perspectives de capitalisation.

Enfin, la cinquième étape (**E5**) consiste à **fournir aux bénéficiaires** (*e.g.* enseignants, apprenants) **les résultats** de l'analyse afin de leur permettre dans une sixième étape (**E6**) d'**utiliser les connaissances produites** pour améliorer l'environnement d'apprentissage et éventuellement effectuer une rétroaction. On peut noter que souvent les bénéficiaires de l'analyse (**R4**) interviennent dès l'étape **E1** en exprimant un besoin qu'ils communiquent au décideur (**R1**). L'étape **E5** de mise à disposition des résultats de l'analyse est une étape délicate, puisqu'il s'agit pour l'analyste de communiquer aux bénéficiaires les connaissances produites lors d'une analyse. Or, bien souvent, ces bénéficiaires ne possèdent pas le même bagage théorique que l'analyste (VOLLE, 2001). Le risque est alors que le bénéficiaire n'interprète et n'utilise pas convenablement ces connaissances, l'amenant ainsi soit à prendre des décisions non pertinentes pour la situation pédagogique étudiée, soit à ne pas agir.

Pour prévenir cela, l'analyste doit donc correctement décrire et organiser la connaissance, que ce soit à destination du bénéficiaire ou à destination d'un système informatique. Pour cela, il est fréquent d'observer des opérations de post-traitement dans l'implémentation d'une analyse, qui permettent de mettre en forme les résultats d'une analyse. De plus, il est important que la communication entre l'analyste et les bénéficiaires soit effective, pour assurer une bonne compréhension des informations transmises. Il est en outre important lors de cette étape de prendre en compte la dimension éthique de l'utilisation qui sera faite des résultats de l'analyse.

L'étape **E6** permet également au bénéficiaire de l'analyse d'interagir avec l'analyste dans le cas où il n'arrive pas à utiliser les résultats. Cet échange peut amener à une reconception de la présentation des connaissances, voire de l'analyse elle-même. Cela peut aussi amener le bénéficiaire à communiquer avec le décideur pour définir une nouvelle formulation du besoin. Durant cette étape, les préoccupations éthiques peuvent également être de nouveau discutées entre le bénéficiaire et le décideur, afin de s'assurer du cadre éthique de la réponse au besoin pour les sujets de l'analyse.

6.1.2 Illustration

Dans cette section, nous illustrons le cycle présenté précédemment sur un exemple que nous reprendrons dans les sections suivantes afin d'illustrer nos propositions. L'exemple est le suivant :

Dans le cadre d'un MOOC, les tuteurs pédagogiques sont confrontés à la problématique de ne pas pouvoir proposer un soutien adapté aux apprenants. Ce faisant, ils voudraient être en mesure d'identifier les apprenants en difficulté et de les aider, ce qui requiert de découvrir des informations pertinentes les concernant dans les différents cours de cette plateforme de MOOC, qui sont dispensés deux fois par an. Les tuteurs, qui jouent le rôle de bénéficiaires (**R4**), communiquent aux responsables (pédagogiques et administratifs) de la plateforme la nécessité d'acquérir des connaissances sur les apprenants pour améliorer l'enseignement. Après avoir affiné les attentes des bénéficiaires, ces responsables – assimilables à nos décideurs (**R1**), formalisent un besoin qui devrait permettre d'aider à améliorer la qualité de l'apprentissage (**E1**) : être en mesure de prédire si un apprenant va, ou non, être certifié à la fin du cours.

Une fois ce besoin clairement défini, l'expert de la plateforme de MOOC (**R2**) procède à l'extraction des traces pertinentes (**E2**). Elles sont anonymisées en accord avec les politiques de confidentialité et d'éthique établies par les décideurs. En outre, il les fournit à l'analyste (**R3**), ainsi que des informations nécessaires pour l'aider dans sa tâche, tout en interagissant avec lui. De telles traces de MOOC sont par exemple accessibles en ligne (MITx et HarvardX, 2014), avec les informations nécessaires à leur compréhension.

Dès lors, l'analyste peut entamer la phase de préparation de l'analyse (**E3**), qui l'amène à échanger avec les décideurs et l'expert pour fixer le type d'analyse à effectuer et prévoir les prétraitements nécessaires. Dans notre cas, il s'agira d'une analyse prédictive : cela découle du besoin qui a été formalisé par le décideur. Ensuite, l'analyste effectue les choix techniques lui permettant de mettre en œuvre les traitements sur les traces (*e.g.* choix de l'outil), avant de concrètement implémenter l'analyse (**E4**).

Un exemple d'implémentation d'une telle analyse est donné par Agnihotric & *al.* (Agnihotri et al., 2016) – accessible en ligne (Agnihotri et al., 2019). Il s'agit d'un tutoriel présenté lors d'un workshop de la conférence EC-TEL 2016 (Kloos et al., 2018), dont les deux premières parties illustrent la réponse à un tel besoin. Cette analyse prédit la certification des étudiants dans un MOOC sur des jeux de données issues de Harvard-MITx, et est réalisée avec le langage de programmation Python. Elle est d'ailleurs décomposable en trois parties, faisant toutes partie de **E4** : certaines concernent le prétraitement des traces, d'autres leur traitement.

La première partie consiste en un prétraitement des données, afin de vérifier les informations communiquées par l'expert (**R2**), et de mettre en forme les traces. Ici, l'analyste crée des variables supplémentaires pour faciliter le traitement des traces, comme la transformation d'une variable "Date of birth" en une variable "Age". Cela fait intervenir de multiples opérations, comme des filtres. La

deuxième partie de l'analyse représente une analyse exploratoire, qui permet d'obtenir des informations supplémentaires sur les traces manipulées. Ici, notamment, la corrélation linéaire pouvant exister entre les propriétés d'un cours et le succès des apprenants est étudiée (en utilisant le coefficient de Pearson). Nous aurons l'occasion de décrire cette étape de corrélation plus en détail dans les chapitres suivants. Enfin, la troisième étape consiste à créer des modèles prédictifs pour répondre au besoin. Pour ce faire, les auteurs exploitent deux types d'approches prédictives, à savoir une approche par régression linéaire, l'autre par catégorisation (*clustering*). Plusieurs modèles sont ainsi proposés avec différentes variables et différentes configurations (*e.g.* nombre de centroïdes différents du cluster), puis éprouvés.

Une fois les modèles jugés pertinents et les résultats obtenus, ils convient qu'ils soient mis en forme puis fournis (**E5**) aux bénéficiaires (**R5**) afin qu'ils puissent les utiliser suivant leur besoin. Par exemple, pour les étudiants dont la prédiction indique qu'ils n'obtiendront pas la certification, les bénéficiaires pourront alors renforcer le suivi pédagogique (**E6**) qui leur est fourni.

6.1.3 Difficultés à réutiliser des analyses

Comme nous l'avons principalement vu dans le Chapitre 3 de l'état de l'art, la difficulté à réutiliser des analyses existantes n'est pas uniquement liée aux contraintes techniques portant sur l'implémentation de l'analyse. Des difficultés peuvent également apparaître lors de la réutilisation d'éléments issus des différentes étapes du cycle d'élaboration de l'analyse, sollicitant les différents acteurs. Dans cette sous-section, nous évoquons certaines de ces difficultés à réutiliser l'existant.

L'étape de formalisation du besoin (**E1**) repose sur l'expertise du décideur pour identifier, de manière la plus efficace possible, les informations nécessaires à la formalisation de ce besoin. Pour préciser la description d'un besoin qui n'est pas encore formalisé, le décideur peut être enclin à consulter diverses analyses déjà mises en œuvre. Cependant, accéder à un recueil de processus d'analyses est à notre connaissance relativement complexe. Bien que quelques outils existent (*e.g.* ^{my}Experiment (C. A. Goble et D. C. De Roure, 2007), UnderTracks (Mandran et al., 2015)), leur utilisation semble marginale au sein de la communauté, d'autant plus qu'ils permettent de ne recueillir que les processus réalisés au sein de chaque outil.

Dès lors, permettre au décideur de rechercher une analyse pouvant correspondre à ses attentes – en ébauche de formalisation – afin de la réutiliser pour l'aider à formaliser son besoin semble revêtir une grande complexité. En effet, l'état de l'art nous montre aussi que l'information fournie par ces outils est souvent textuelle et peu structurée (cf. Chapitre 4). Or, cela ne contribue pas à faciliter la recherche, puisqu'une telle information est difficilement exploitable par la machine. En outre, pour un décideur, identifier à quel besoin répond une analyse nécessite une expertise dans l'analyse de traces : par conséquent, cela affecte aussi négativement l'identification d'une telle analyse comme réponse au besoin actuel du décideur. Quand bien même, il se pose au final pour le décideur, le problème de la compréhension des analyses décrites dans ces outils. Malgré une approche exploratoire (Mandran et al., 2015), les analyses stockées dans ces outils ont un haut niveau de technicité et présentent peu d'informations (*e.g.* contextes, retour utilisateurs, contraintes). Le besoin traité est souvent relayé au second plan, ne proposant que l'implémentation de l'analyse. Ainsi, un décideur ne pourra que difficilement être apte à réutiliser correctement une analyse déjà existante pour formaliser son besoin.

Lors de la réutilisation d'une analyse déjà implémentée, des difficultés importantes peuvent également survenir lors de la phase de sélection des traces pertinentes (**E2**). En effet, l'analyse qui est réutilisée a été réalisée pour un cas précis : elle est fondée sur des traces spécifiques à cette situation. La tâche de l'expert (**R2**) est alors d'être capable de faire l'analogie entre les traces de l'analyse réutilisée et sa situation actuelle. En plus de devoir comprendre les traces de l'analyse réutilisée pour permettre leur analogie avec d'autres variables, il doit identifier quelles variables sont importantes, et comment les adapter.

À ce moment, des mises en correspondance incorrectes peuvent survenir. Bien que ce type d'erreur soit susceptible d'être détecté lors de la phase de préparation et d'implémentation de l'analyse, ces erreurs

peuvent altérer la pertinence des résultats du futur processus d'analyse, voire le processus lui-même. De plus, pour s'assurer que les traces qu'il sélectionne sont pertinentes, l'expert doit aussi identifier les spécificités de l'analyse, comme le contexte pédagogique, et en tenir compte pour respecter les contraintes de l'analyse réutilisée (par exemple la taille d'un échantillon pour une classification). Il se confronte donc lui aussi à une approche exploratoire de l'analyse et des données qui y sont associées de manière éparse.

Concernant la phase de préparation de l'analyse (**E3**) menée par l'analyste (**R3**), avoir recours à des analyses existantes constitue un atout majeur. Cela permet à l'analyste d'avoir accès à des méthodes d'analyse spécifiques au besoin auquel il cherche à répondre. Pour chaque analyse existante examinée, l'analyste doit d'abord comprendre cette analyse puis la comparer avec le besoin auquel il doit répondre et les traces dont il dispose pour concevoir son analyse. Il peut ainsi trouver une analyse existante qui répond exactement à son besoin et qui pourra s'appliquer sur ses traces. Si ce n'est pas le cas, il peut aussi trouver des parties d'analyses existantes qui sont pertinentes par rapport à son besoin. Pour réutiliser une analyse ou une partie d'analyse, l'analyste doit avoir accès à l'outil avec lequel elle a été implémentée, comprendre les opérateurs utilisés ainsi que leur paramétrage.

La description de la manière dont les analyses sont implémentées est donc importante pour l'analyste. Or, les outils d'analyse existants sont destinés à rendre les analyses exécutables, mais les informations associées au choix d'élaboration et d'implémentation sont rarement décrites, et peu structurées. Elles sont de plus spécifiques à l'outil utilisé. Cela complexifie la tâche de l'analyste lorsqu'il doit, par exemple, identifier le type de l'analyse existante (*e.g.* prédictive ou diagnostique), ou encore choisir des étapes pertinentes d'une analyse qu'il souhaite réutiliser : il faut par exemple exclure les étapes de prétraitements spécifiques aux traces considérées dans l'analyse existante. De plus, sans la possibilité d'avoir accès aux différents retours des acteurs, l'analyste prend le risque de réutiliser une analyse erronée ou caduque. Les retours de chercheurs permettraient par exemple d'attester la pertinence scientifique de l'analyse réutilisée.

Concernant l'implémentation de l'analyse (**E4**), réutiliser une analyse existante implémentée dans un autre outil que celui qu'utilise habituellement l'analyste est actuellement complexe (Belhajjame et al., 2015). Il peut aussi arriver que l'analyste souhaite réutiliser plusieurs parties d'analyses implémentées dans différents outils. Or, comme nous l'avons vu, les outils d'analyse possèdent leur propre formalisation des processus d'analyse. Par exemple, R adopte une approche programmatique (R Development Core Team, 2008) alors que UnderTracks (Mandran et al., 2015) adopte une approche fondée sur les workflows. En outre, les outils d'analyse possèdent généralement leur propre moteur d'exécution. Ainsi, un opérateur développé dans un outil ne pourra pas être réutilisé tel quel dans un autre outil d'analyse et nécessitera, au mieux, d'être adapté. Par exemple, deux opérateurs similaires dans deux outils attendent des paramétrages différents.

Par conséquent, deux analyses répondant à un même besoin seront implémentées différemment au sein de deux outils différents. Dès lors, l'analyste qui essaie de réutiliser une analyse existante est confronté d'une part à la nécessité de l'importer dans l'outil qu'il a décidé d'utiliser (*e.g.* son outil de prédilection), et d'autre part de l'adapter, en modifiant les opérateurs incompatibles, leur paramétrage et l'enchaînement des étapes lorsque cela ne convient plus. Si ce n'est pas possible, il doit recréer l'analyse entièrement – une tâche difficile et potentiellement source d'erreurs.

Pour réutiliser une analyse existante, en plus de ces difficultés liées aux contraintes techniques, l'analyste doit également tenir compte des modifications qu'implique le contexte d'analyse dans lequel il se trouve. Là encore, les informations associées à l'analyse réutilisée sont capitales pour permettre à l'analyste de savoir comment la modifier et la rendre pertinente pour son contexte. Mais, comme dit précédemment, ces informations sont peu présentes et peu structurées dans les outils d'analyse actuels. Cela implique que l'analyste peut n'avoir aucun moyen pour évaluer la pertinence de l'analyse réutilisée (*e.g.* théories utilisées, configurations, jeux de tests), et donc des résultats obtenus.

Ainsi, nous le voyons bien, permettre de réutiliser des analyses existantes représente un enjeu majeur lors de toutes les étapes de l'élaboration d'une analyse. Cela apporte des supports à l'élaboration, des éléments constructifs et peut améliorer la pertinence de l'analyse en cours de création. Cependant, la réutilisation et l'adaptation des processus d'analyse sont entravées par la difficulté d'accéder à de

telles analyses, par les contraintes techniques occasionnées par les outils, et par les contraintes liées au contexte de l'analyse. De plus, le manque d'information sur les analyses existantes et la faible structuration de l'information présente ont également un impact non négligeable.

Ces raisons contribuent à faire émerger la nécessité de proposer un moyen de représenter les processus d'analyse qui permettrait de les réutiliser dans d'autres situations pédagogiques, en tenant compte de leurs spécificités et des choix d'implémentation. Une telle représentation devra faciliter la tâche de réutilisation d'une analyse existante pour chacun des acteurs impliqués dans les différentes étapes du cycle d'élaboration d'un processus d'analyse. Cela nécessite en particulier de structurer les informations représentant :

— le besoin auquel répond le processus d'analyse ;
— les contraintes sur les traces qu'il permet de traiter ;
— les contraintes techniques sur les opérateurs d'analyse utilisés ;
— les contraintes liées au contexte pédagogique pour lequel les résultats du processus d'analyse sont pertinents.

Notre objectif est ainsi de capitaliser les processus d'analyse de traces d'apprentissage, tout en considérant les différents acteurs comme une part intégrante de leur mise en œuvre.

6.2 La capitalisation et ses propriétés intrinsèques

L'étude de la littérature concernant les traces d'apprentissage, les processus d'analyse de traces et les outils d'analyse fait apparaître un tropisme de partage lorsque l'objectif est de pouvoir réutiliser les analyses. En effet, il semble de prime abord naturel que, pour réutiliser quelque chose, l'on doive avant tout la partager. Néanmoins, avoir un processus qu'il est possible de s'échanger de pair à pair n'induit en rien la possibilité de pouvoir le réexploiter (Belhajjame et al., 2015). De plus, les travaux sur l'interopérabilité, comme ceux de PMML (Data Mining Group, 2018[j]), essaient certes de répondre aux problématiques techniques du partage des processus d'analyse, mais n'en garantissent pas non plus leur cohérence ou leur maintien (*e.g.* cohérence scientifique, pédagogique).

En outre, la notion même de réutilisation de processus d'analyse est vague puisqu'elle englobe plusieurs aspects liés aux différentes étapes de conception du cycle d'analyse et aux différents acteurs impliqués (cf. Section précédente). En plus de cela, on y décèle une polysémie qu'il est nécessaire de clarifier pour pouvoir définir la capitalisation, et ainsi aborder correctement le problème de la capitalisation.

En effet il est important de définir clairement cette propriété de capitalisation. Dès lors, comme nous l'observons ci-après, elle est constituée de différentes propriétés qui se combinent, dont certaines pouvaient couramment être considérées comme équivalentes.

Nous spécifions ici la capitalisation, notamment en clarifiant le concept de réutilisation et en identifiant les propriétés qui sont sous-entendues par cette notion de réutilisation d'un processus d'analyse. Nous montrons également comment ces différentes propriétés se combinent pour permettre la capitalisation.

L'ambiguïté du terme de réutilisation se remarque dans les différents travaux (Springer Nature, 2016 ; Bánáti et al., 2015) portant sur une démarche scientifique ouverte et reproductible, comme ceux du Vocabulaire International de Métrologie (Guides in Metrology, 2008) ou ceux d'ACM (Association for Computing Machinery) (ACM, 2017). Dans ces travaux, trois termes émergent comme synonymes à celui de "réutilisation", à savoir la reproductibilité, la réplicabilité et la répétabilité.

Néanmoins, nous avons identifié dans ces travaux trois questions principales qui servent à caractériser la réutilisabilité d'un élément :

— L'élément réutilisé doit-il l'être par les personnes qui l'ont initialement conçu ?

— L'élément peut-il être utilisé dans un contexte (*e.g.* technique, pédagogique, expérimental) différent du contexte original ?
— L'élément peut-il utiliser des ressources (*e.g.* traces) similaires à celles utilisées à l'origine ?

De plus, la définition des termes de réplicabilité, de répétabilité, de réutilisation et de reproductibilité semble réunir un meilleur consensus dans certaines disciplines, comme celle des workflows (BÁNÁTI et al., 2015 ; GOECKS et al., 2010). C'est pourquoi nous avons décidé de nous appuyer sur ces travaux pour définir ce que nous considérons comme la capitalisation. Nous nous servons également des questions précédentes pour discriminer plus efficacement les différents termes inhérents à notre définition de la capitalisation, et pour montrer que cette capitalisation est un ensemble interdépendant de différentes propriétés.

La **réplicabilité** d'un processus d'analyse signifie que les opérateurs impliqués dans ce processus sont clairement identifiés et que leur ordre d'application est défini. Dès lors, la réplication ne nécessite aucune description du contexte et peut être considérée comme la simple succession ordonnée d'opérateurs d'une analyse, indépendamment des résultats qu'ils produisent. Il s'agit alors de sa définition sur le plan structurel. Entendez qu'un processus est réplicable s'il est possible d'en connaître tous les opérateurs qui ont été utilisés pour obtenir un résultat quelconque.

La **répétabilité** d'un processus d'analyse fait intervenir la traçabilité des résultats produits et leur constance vis-à-vis d'un même jeu de données. Cette immuabilité se manifeste principalement dans la possibilité de traduire le déterminisme des opérations utilisées dans le processus d'analyse [1] (BRECKENRIDGE, 1989 ; KAMBATLA et al., 2014 ; IHANTOLA et al., 2015). Aussi, les paramètres des opérateurs et leurs effets doivent être clairement définis, et les modèles utilisés (*e.g.* régression linéaire) doivent être communiqués ou, à défaut, doivent avoir l'ensemble des données d'entraînement disponible pour les reproduire à l'identique. Autrement dit, lorsqu'une analyse est de nouveau exécutée avec les mêmes configurations, les mêmes phénomènes doivent y être observés (*e.g.* même résultats en sortie) [2]. La répétabilité permet donc de vérifier les résultats produits lors d'une analyse et ainsi d'identifier d'éventuels biais scientifiques ou techniques. Elle est une composante nécessaire pour une science ouverte et accessible.

La **réutilisabilité** en tant que telle signifie qu'une analyse est conçue de manière à pouvoir être utilisée avec d'autres jeux de traces que ceux utilisés initialement, pour peu que ces traces soient analogues (*i.e.* même contexte pédagogique) et que les variations du contexte technique soient nulles, voire négligeables. Ainsi, sur des données similaires, seules des modifications mineures de l'implémentation de l'analyse seront nécessaires. Ces modifications mineures n'altèreront pas les théories scientifiques sur lesquelles s'appuie l'analyse initiale, ce qui garantit la pertinence des résultats obtenus.

La Figure 6.2 représente la manière dont ces trois propriétés s'articulent : pour être réutilisable, une analyse doit être répétable. Et pour être répétable, elle doit être réplicable. Autrement dit, pour permettre une nouvelle exécution, les opérateurs et leur ordre doivent être préalablement connus. Ces trois propriétés permettent la **reproductibilité** d'un processus d'analyse. Cependant, selon nous, la reproductibilité ne permet pas de répondre à elle seule à toutes les problématiques liées à la capitalisation qui ont été soulevées préalablement lors de l'état de l'art (*e.g.* manque d'informations supplémentaires), ce qui n'en fait pas la propriété de capitalisation en elle-même, mais une de ses composantes. En effet, au vu de notre étude de la littérature, la capitalisation des processus d'analyse requiert trois propriétés supplémentaires : la compréhension, l'ouverture et enfin l'adaptation.

Une analyse est **compréhensible** si les différents aspects de l'analyse sont appréhendables par les acteurs concernés. Pour cela, il faut décrire les informations techniques, mais aussi les informations conceptuelles, comme les objectifs de l'analyse, les théories scientifiques utilisées, ou encore les choix d'implémentation. Or, nous l'avons vu, les acteurs impliqués dans l'élaboration et l'exploitation des processus d'analyse possèdent des expertises variées. La conséquence directe est une impossibilité potentielle à comprendre des parties du processus. Ainsi, pour que ces acteurs puissent réutiliser un

1. Nous ne considérons pas ici l'architecture matérielle. Toutefois, dans le cas de systèmes distribués ou fortement décentralisés, il serait pertinent de s'y intéresser également.
2. Cela fait d'ailleurs intervenir les problèmes d'évolution des outils d'analyse et des services de calculs distants : d'une version à l'autre, la manière de calculer peut changer drastiquement (*e.g.* changement de format, normalisation des résultats).

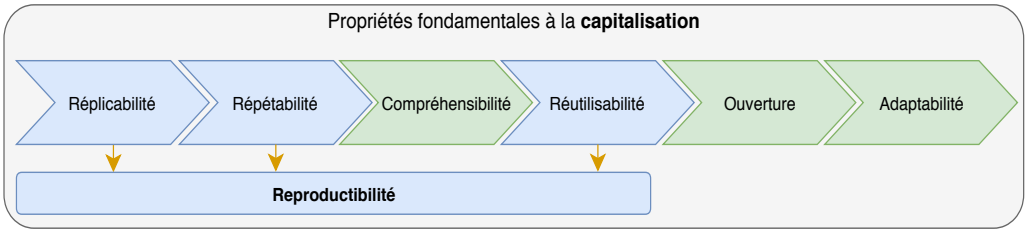


Figure 6.2.: Illustration des propriétés requises pour la capitalisation des processus d'analyse de traces. L'imbrication des flèches indique la dépendance entre les propriétés.

processus d'analyse existant sur des traces différentes ou modifier un élément de ce processus, il est nécessaire qu'au préalable ce processus soit correctement décrit à chaque étape de son élaboration. Il faut aussi rappeler que le manque de compréhension des analyses constitue une des principales raisons qui empêchent leur réutilisation (BELHAJJAME et al., 2015).

Rendre une analyse **ouverte** requiert de rendre le processus consultable pour tout un chacun, par exemple en le déposant sur un dépôt libre et accessible à tous, ce qui induit la notion de partage. Dans cette notion d'ouverture, nous intégrons de plus la notion de portabilité : le processus doit conserver sa cohérence scientifique et technique lorsqu'il est déployé sur un autre outil d'analyse. Ceci nécessite que le processus d'analyse (ainsi que ses composants) soit décrit indépendamment de l'outil d'analyse dans lequel il a été implémenté.

Enfin, à la différence de la réutilisabilité, l'**adaptabilité** d'un processus d'analyse signifie que des modifications peuvent être réalisées pour appliquer le processus d'analyse sur d'autres traces – potentiellement différentes – ou pour répondre à des besoins d'analyse différents. Cependant, ces nouveaux besoins doivent tout de même se situer dans des contextes cohérents à ceux de l'analyse initiale, afin de respecter les fondements théoriques de cette analyse et ainsi éviter des résultats non pertinents.

La Figure 6.2 montre comment ces trois propriétés s'articulent avec la notion de reproductibilité pour définir la capitalisation. Sur cette figure, chaque propriété est requise par celle qui lui succède. Nous pouvons donc en conclure que considérer le partage comme le socle de la capitalisation ne représente pas une approche viable. Sans l'une des six propriétés susmentionnées, nous estimons que la capitalisation effective d'un processus d'analyse n'est pas possible.

6.3 Notre approche pour capitaliser

Actuellement, si l'un des acteurs de l'analyse souhaite réutiliser des analyses pour pourvoir à ses besoins, il doit d'abord être en mesure d'en rechercher qui puissent convenir. Il se confronte aux nombreuses difficultés que nous avons illustrées dans la Section 6.1.3, qui sont liées au fait que les processus d'analyse dépendent toujours du contexte technique et pédagogique dans lequel ils ont été réalisés. Par exemple, l'absence d'entrepôt commun de processus techniquement indépendants qui l'oblige à déceler les ingérences du contexte technique pour l'adapter. Mais aussi les possibilités de recherche qui sont à sa disposition (*e.g.* moteur de recherche, articles scientifiques) qui peuvent ne pas être adaptées.

Cette recherche se fait sans aucune assurance, d'une part, que l'acteur l'opère correctement et, d'autre part, que les éventuels résultats soient valides pour les besoins de l'acteur. Combiné avec la spécialisation technique des processus induite par l'outil utilisé et la situation pédagogique sur laquelle ils portent, cela contribue à créer un contexte où développer de zéro un processus d'analyse – sans forcément consulter l'existant – s'avère moins complexe que comprendre l'existant et l'adapter pour le réutiliser aux dépens de la perte que cela représente. L'existence du projet HUBBLE (PROJET HUBBLE, 2016) et de certaines initiatives (LEARNSPHERE, 2018[d] ; APEREO FOUNDATION, 2016) sont d'ailleurs la preuve de la prise de conscience de la communauté de l'importance d'être capable de réutiliser l'existant, et non plus de le développer à chaque fois. Le cycle d'élaboration et d'utilisation

(cf. Figure 6.1) que nous avons présenté illustre ce constat : il est hermétique à l'existant et il n'y a aucune relation explicite entre l'existant et les acteurs ou les étapes.

Nous proposons un nouveau paradigme des processus d'analyse de traces d'apprentissage. Ce paradigme consiste à affranchir les processus d'analyse, ainsi que leurs opérateurs et les traces utilisées, des contraintes techniques liées aux outils d'analyse, en les rendant indépendants des outils d'analyse. Pour cela, nous tâchons de les formaliser en remontant jusqu'au concept qu'ils représentent, et qui sont primairement manipulés par l'analyste – avant qu'ils ne soient impactés par les contraintes techniques. Nous montrons grâce à nos résultats expérimentaux que l'indépendance technique est une première étape nécessaire dans la capitalisation.

Avec ce paradigme, nous intégrons également directement dans les processus d'analyse une description structurée de leurs informations relatives, afin de faciliter leur compréhension, adaptation et réutilisation. Notre approche a également pour objectif de permettre l'implication des différents acteurs intervenant dans la conception d'un processus d'analyse, pour permettre une co-construction de ces processus. Pour cela, nous adoptons une approche narrative pour modéliser les processus d'analyse. Nos résultats expérimentaux (cf. section 14, page 179) montrent que notre approche narrative complète l'indépendance technique, et peut répondre à la problématique de la capitalisation au sein de notre communauté.

Dès lors, il devient possible de faire cohabiter ces processus d'analyse au sein d'un même dispositif de stockage, et donc de faire émerger des relations entre eux. Il en résulte alors la possibilité d'un entrepôt à destination des différents analystes, où différents mécanismes d'assistance peuvent être déployés pour correspondre aux besoins et aux différentes expertises des acteurs. La Figure 6.3 schématise l'ensemble de notre approche.

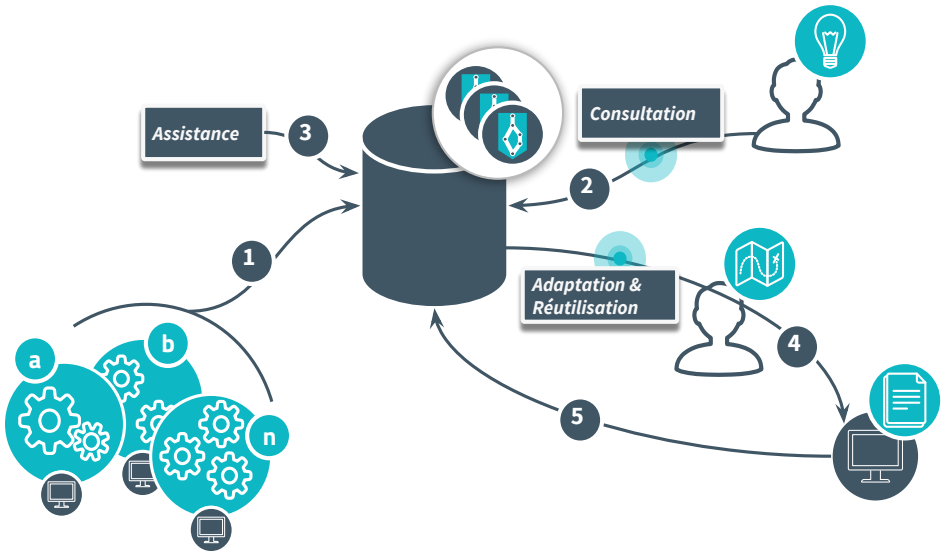


Figure 6.3.: Illustration de l'approche. Des processus d'analyse existants sont transformés dans notre formalisme indépendant, et enrichi par les différents acteurs (1). Ils peuvent être consultés (2) par les différents acteurs en fonction de leurs attentes, et ces acteurs aidés par des assistances automatiques (3). Enfin, ces processus sont réutilisés (4) et injectés de nouveau (5) pour alimenter la diversité des processus disponibles.

Nous défendons donc l'hypothèse que l'analyse de traces est formalisée dans son état primordial indépendamment des contraintes techniques et que les contraintes pédagogiques et leurs influences y sont clairement identifiables. Une telle analyse est donc instanciable dans différents outils, est adaptable pour des besoins d'analyse différents. Pour étayer cette thèse, il faut observer l'analyse de traces d'apprentissage dans son cadre phénoménologique. Husserl (Husserl, 1950) introduit la notion

d'objet intentionnel [3] de l'acte de penser, le **noème**, et le processus conscient du travail cérébral sur l'objet – l'acte de penser, la **noèse**.

Ainsi, il est possible d'assimiler le besoin d'analyse et les traces à des noèmes, qui sont perçus d'une manière singulière par les différents acteurs. Les différentes interactions qui surviennent alors durant l'élaboration de l'analyse constituent autant de modificateurs de perception pour ces acteurs. Enfin, l'analyse en elle-même, avant son implémentation dans un outil d'analyse, est représentative de la noèse opérée par l'analyste. De plus, Rosh (ROSCH, 1973) formule le fait que la cognition s'effectue par l'utilisation de catégories d'objets qui font office de "point de référence cognitif", plutôt que par des instances élémentaires de ces objets.

Dès lors, il apparaît possible pour un analyste d'identifier les catégories d'objets qu'il a manipulés lors de la mise en œuvre de l'analyse, et de les faire correspondre à des concepts abstraits, eux-mêmes définis indépendamment du contexte technique. Un tel processus ne souffrirait alors pas des contraintes liées à son implémentation dans les outils d'analyse. De plus, cela offrirait la possibilité de s'adapter à différentes instances des traces utilisées et du contexte pédagogique.

Pour terminer ce chapitre, nous illustrons l'impact de la capitalisation des processus d'analyse de traces d'apprentissage sur le cycle d'élaboration et d'exploitation des analyses. La Figure 6.4 l'illustre.

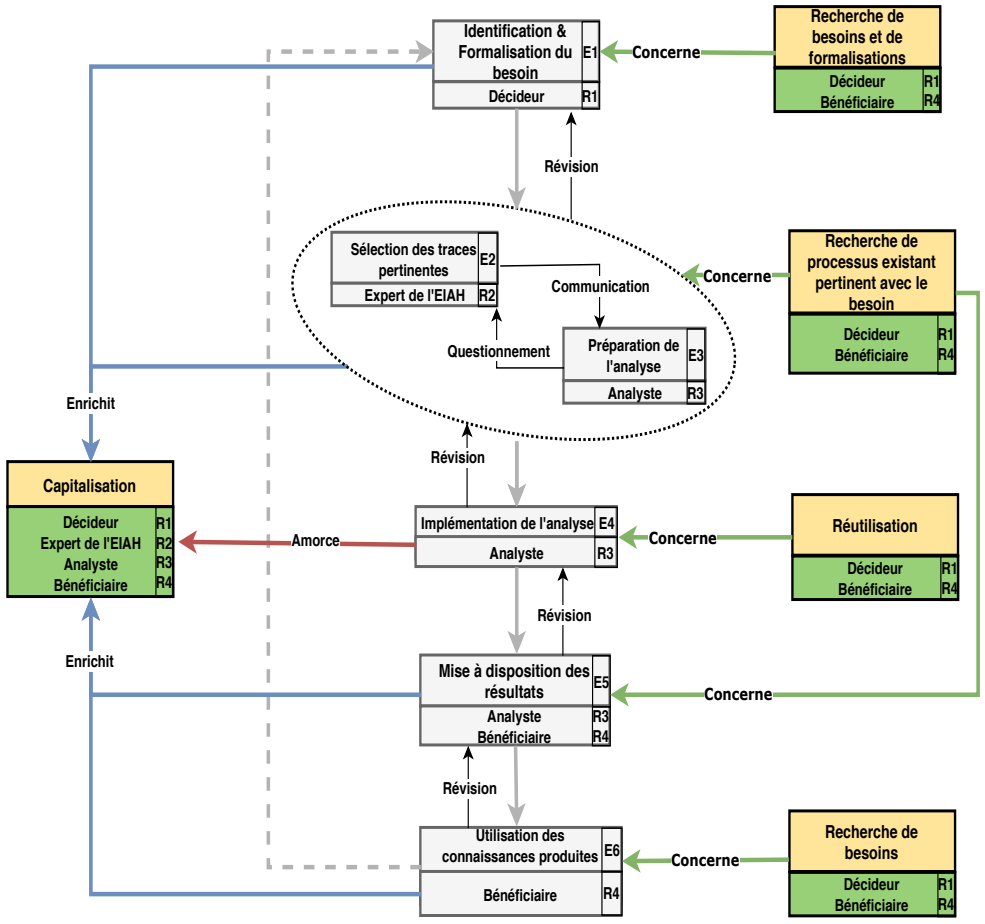


Figure 6.4.: Illustration de l'intégration de la capitalisation dans le cycle d'élaboration et d'exploitation d'un processus d'analyse.

Nous considérons que la capitalisation d'un processus d'analyse dans son ensemble (*i.e.* le processus en lui-même et aussi les informations associées) s'amorce à l'initiative de l'analyste. En charge de la mise

3. Il ne s'agit pas de l'objet (*i.e.* un arbre) tel qu'il peut s'établir dans son être réel (*i.e.* la réalité), mais bien tel qu'il nous apparaît, d'après nos expériences, sensibilités, *etc*.

en œuvre de l'analyse, et en interaction avec tous les autres acteurs du cycle, il possède en effet la vision la plus globale de la réponse à un besoin d'analyse. Ce constat s'est, de plus, observé dans le cadre du projet Hubble (PROJET HUBBLE, 2016 ; PROJET HUBBLE, 2018). Néanmoins, ce rôle n'évince pas les autres acteurs dans l'enrichissement du processus d'analyse ainsi capitalisé, pour fournir des informations capitales, comme les choix réalisés, les questions répondues, les traces sélectionnées ou encore les hypothèses posées. La partie gauche de la Figure 6.4 illustre tout cela.

Il est intéressant d'observer les différents apports issus de la capitalisation dans le cycle d'élaboration et d'exploitation de l'analyse. Nous en présentons des majeurs dans la partie droite de la Figure 6.4, qui concernent les différentes étapes du cycle. Dans le cas de l'implémentation de l'analyse d'abord (partie centrale de la figure), il devient alors possible pour l'analyste de réutiliser l'existant pour répondre au besoin d'analyse (cf. rectangle *Réutilisation* de la partie droite).

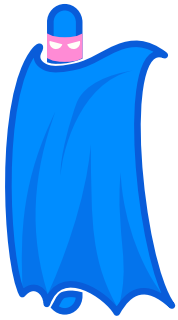
En outre, cela va influer sur toutes les autres étapes du cycle. Lors de la préparation de l'analyse (E3), cette mise à disposition de l'existant sert également l'analyste puisqu'elle lui permet de rechercher des processus analogues, et de s'en inspirer. Il peut par exemple se documenter sur les techniques à employer et la manière de les exploiter. De plus, cela peut permettre de renforcer l'échange entre cette étape de préparation et celle de sélection des analyses, en s'appuyant sur des analyses concrètes (cf. rectangle *Recherche de processus*). Lorsque le processus est mis à disposition des bénéficiaires (E5) par exemple, l'analyste peut rechercher dans les processus connexes à celui qu'il a implémenté quelles ont été les meilleures manières de présenter et communiquer les résultats (*e.g.* mettre en forme des catégories d'apprenants) : l'on voit ici l'importance d'enrichir un processus capitalisé avec des informations supplémentaires, comme les hypothèses posées ou encore les relations existantes entre les différents éléments de l'analyse. L'avantage ici est que les bénéficiaires peuvent aussi exprimer leurs préférences en s'appuyant eux aussi sur l'existant – intrinsèquement compréhensible.

Cet accès à l'existant permet également aux bénéficiaires une approche exploratoire sur les besoins qu'il est possible de résoudre (cf. rectangle *Recherche de besoins*), suggérant ainsi une certaine sérendipité dans la démarche. Cela vaut également pour le décideur lorsqu'il tente de formaliser le besoin d'analyse (E1), puisqu'il est alors en mesure de s'appuyer sur l'existant pour renforcer sa formalisation, voire décider de réutiliser une analyse déjà existante pour répondre au besoin (cf. rectangle *Recherche de besoins formalisés*). Enfin, pour revenir sur la Recherche de processus, l'expert des EIAH est également concerné. Il pourra identifier les traces qui sont nécessaires pour répondre à un besoin d'analyse spécifique, et s'en servir pour faire l'analogie avec les traces que lui possède dans son environnement.

Finalement, cette partie nous permet de mettre en avant que les informations fournies par les acteurs du cycle d'élaboration et d'exploitation de l'analyse constituent un socle à la compréhension et à la réutilisation des analyses. La capitalisation, pour exister, doit alors pouvoir tenir compte de cette interdépendance entre les acteurs, en plus des caractéristiques de l'analyse elle-même.

Abstraire les processus d'analyse de traces

7



Sommaire

Section	**Introduction**	**93**
Section 7.1	**Description de l'approche**	**94**
Section 7.2	**Méthodologie de création des méta-modèles**	**96**
Section 7.3	**Formalisation du méta-modèle CAPTEN-ALLELE**	**97**
7.3.1	Méta-modèle d'une variable	97
7.3.2	Méta-modèle d'une liste	98
7.3.3	Méta-modèle d'un opérateur indépendant	99
7.3.4	Méta-modèle d'un processus d'analyse indépendant	100
7.3.5	Fonctionnement et grammaire	101
Section 7.4	**Illustration**	**103**
Section	**Ce qu'il faut retenir**	**105**

Publications relatives à ce chapitre

(Lebis et al., 2016) A. Lebis, M. Lefevre, V. Luengo et N. Guin (2016). "Towards a Capitalization of Processes Analyzing Learning Interaction Traces". In : *11th European Conference on Technology Enhanced Learning (EC-TEL 2016)*. T. 9891. Lecture Notes in Computer Science. Lyon, France : Springer, p. 397–403

(Lebis, 2016) A. Lebis (2016). "Vers une capitalisation des processus d'analyse de traces". In : *Rencontres Jeunes Chercheurs en EIAH (RJC-EIAH 2016)*. Montpellier, France

(Lebis et al., 2017b) A. Lebis, M. Lefevre, V. Luengo et N. Guin (2017b). "Capitaliser les processus d'analyse de traces d'e-learning". In : *Méthodologies et outils pour le recueil, l'analyse et la visualisation des traces d'interaction - ORPHEE-RDV*. Font-Romeu, France

Introduction

L'état de l'art vu précédemment (cf. Partie I), ainsi que la spécification des propriétés de la capitalisation (cf. Chapitre 6, Section 6.2), nous conduisent à identifier les contraintes techniques comme le premier obstacle à dépasser pour viser la capitalisation. En effet, ces contraintes qui sont induites par de

nombreux facteurs, comme les outils d'analyse et leur nécessité d'exécuter des opérations, contraignent les processus d'analyse et la manière de les mettre en œuvre, de telle sorte qu'il devient complexe de pouvoir les réutiliser de nouveau (BELHAJJAME et al., 2015).

Or, ce contexte technique des analyses varie et il n'existe actuellement pas de moyen d'en tenir compte et de l'expliciter. Nous en voulons pour exemple la présence de paramétrages supplémentaires pour un opérateur de *clustering* dans R comparé à celui de RapidMiner : il revient à l'analyste la charge d'identifier les différences et de les interpréter pour choisir l'outil le plus approprié, afin qu'il adapte la mise en œuvre de l'analyse à l'outil.

Dans ce chapitre, nous présentons notre proposition qui vise à affranchir les processus d'analyse des contraintes techniques liées aux outils d'analyse et aux traces. Il s'agit d'en proposer une modélisation plus abstraite grâce à la proposition de méta-modèles. Aussi, nous décrivons dans la Section 7.1 notre approche, avant de détailler dans la Section 7.2 la méthodologie suivie pour établir ces méta-modèles. Dans la Section 7.3, nous formalisons les méta-modèles, mais aussi la manière dont ils sont utilisés pour construire l'analyse. Puis, nous illustrons l'utilisation de notre proposition dans la Section 7.4.

La mise en œuvre de cette proposition et les résultats expérimentaux associés sont présentés respectivement dans la Partie III, Chapitre 10 et dans la Partie IV, Chapitre 13. Cette organisation du manuscrit nous permet de faire suivre cette abstraction des processus d'analyse par la modélisation des éléments supplémentaires nécessaires à la capitalisation (cf. Chapitre 8).

7.1 Description de l'approche

En prémisse de nos travaux, il est important de noter qu'un processus se compose comme une séquence ordonnée d'opérations. Dans sa forme la plus simple, l'on peut le formaliser de la manière suivante :

Définition 1.1 (Processus d'analyse)

Dans sa forme minimale, un processus d'analyse $\mathcal{A}$ se définit par le tuple :

$$\mathcal{A} = \langle \mathcal{T}, \mathcal{O}, E, \gamma, < \rangle \tag{7.1}$$

Où $\mathcal{T}$ représente les traces utilisées et $\mathcal{O}$ les opérateurs.
E représente comment les opérateurs sont reliés – liens assimilables à des arcs dans un graphe, $\gamma : E \rightarrow \mathcal{O} \times \mathcal{O}$ une fonction qui associe à chaque arc une paire d'opérateurs, et $<$ une relation d'ordre partiel sur les opérateurs.

Pour qualifier un processus d'analyse d'indépendant techniquement, il convient donc d'attendre que tous ses composants puissent être qualifiés comme tels, notamment les opérateurs et les traces manipulées.

Pour cela, nous proposons d'introduire des méta-modèles permettant de décrire les **concepts** mis en œuvre dans les différentes étapes d'un processus d'analyse. En cela, notre approche trouve écho dans la réflexion de Rosch (ROSCH, 1973) au sujet de la cognition qui s'effectue par catégories d'éléments plutôt que par des instances élémentaires de ses objets. En effet, nous considérons qu'un opérateur véhicule un concept d'opération précis, en plus de ses attributs techniques, et que c'est ce concept qui motive son utilisation. Il en va de même pour les traces. Par exemple, il réside dans les opérations dites de corrélation l'objectif concret d'extraire le lien existant entre deux variables, et ceci, indépendamment de la manière dont ils sont mis en œuvre dans les outils d'analyse.

L'hypothèse de notre approche est de supposer qu'il est possible de capturer au sein d'une modélisation cette partie conceptuelle des opérateurs et des traces. Isolés des contraintes techniques qui sont spécifiques au contexte de l'implémentation, il devient alors possible de les représenter de manière unifiée.

Cette représentation unique fait donc opposition à la manière actuelle dont sont définis ces concepts dans les outils d'analyses : chaque concept n'est potentiellement disponible que dans un sous-ensemble d'outils d'analyse et implémenté différemment (*i.e.* l'opérateur), du fait des spécificités techniques de chaque outil. Nous avons ainsi choisi de privilégier un niveau d'abstraction supplémentaire pour la description des processus d'analyse, en acceptant que les processus ainsi décrits ne soient pas directement exécutables. Nous sommes donc en présence d'une approche descendante [1], où le processus se voit décrit indépendamment des spécificités techniques, pour ensuite être instancié dans les outils d'analyse usuels.

Les méta-modèles que nous proposons pour constituer cette approche traduisent les notions primordiales de traces, d'opérateurs et de processus d'analyse dans un caractère indépendant, comme nous le présentons ci-après.

Nous définissons une **variable** comme une représentation abstraite d'un élément de trace. Il s'agit d'expliciter les concepts présents dans les traces, en se dégageant des valeurs présentes dans les traces et de leur format. Par exemple, dans un fichier de traces au format CSV, il s'agit de décrire ce que représente une colonne de données (*e.g.* une durée), sans tenir compte des valeurs contenues dans cette colonne (la durée pouvant être exprimée dans plusieurs unités de temps). Nous rejoignons ainsi les travaux de Rosh (ROSCH, 1973).

De la même manière, nous définissons un **opérateur indépendant** comme la représentation abstraite d'un ensemble d'opérateurs similaires partageant le même concept d'opération : ils ont la même sémantique. Dans l'exemple de la Section 6.1.2 ci-avant, les opérateurs de corrélation peuvent être trouvés dans divers outils d'analyse, et l'opérateur indépendant "corrélation" $o_{correlation}$ en représente l'objectif qu'ils ont en commun : extraire un lien entre des variables.

Néanmoins, même si nous avons choisi un cadre de modélisation plus abstrait nous privant de fait de la possibilité d'exécuter les opérateurs directement, il n'en est rien en ce qui concerne l'interprétation de l'impact qu'ils auront sur les variables. En effet, la transition d'état des traces effectuée par un opérateur, c'est-à-dire la manière dont les variables fournies en entrée ont évolué en sortie, peut être représentée dans notre approche. Il est ainsi possible de décrire le processus jusque dans l'évolution des variables au cours de l'analyse.

Pour ce faire, un opérateur indépendant utilise des variables d'entrée et produit des variables de sortie. Il est également possible d'indiquer les paramètres d'un opérateur indépendant : ils sont identifiés à partir de l'ensemble des paramètres communs à tous les opérateurs concernés par cet opérateur indépendant. Dans le cas où un opérateur est amené à produire comme résultat un modèle, il est alors directement envisageable de définir l'opérateur indépendant associé.

La Figure 7.1 illustre l'opérateur indépendant "corrélation" $o_{correlation}$, qui représente le dénominateur commun à des opérateurs de corrélation implémentés dans différents outils d'analyse X, Y et Z. Cet opérateur indépendant représente l'ajout d'une variable "corrélation" décrivant comment les variables V_1 et V_2 sont corrélées, mais ne permet pas, du fait de son niveau d'abstraction, d'effectuer le calcul de la corrélation.

Enfin, nous définissons un **processus d'analyse indépendant** comme une succession d'opérateurs indépendants. Le premier opérateur est appliqué sur des variables dites initiales qui représentent alors les concepts originaux, ceux qui sont contenus dans une trace. Les variables de sortie de chaque opérateur peuvent ensuite être utilisées comme variables d'entrée par l'opérateur suivant ; elles perdent alors leur statut de variable initiale. Cela permet de modéliser, toujours à l'aide d'entités abstraites, la séquence d'opérations nécessaires et les variables qui en résultent.

Le fait que les opérateurs utilisés par le processus d'analyse indépendant soient eux-mêmes indépendants des contraintes techniques des outils d'analyse, et que les variables utilisées par ces opérateurs soient également indépendantes des formats des traces, nous assure l'indépendance des processus d'analyse ainsi construits. Un processus d'analyse indépendant permet ainsi de représenter la mé-

1. top-down

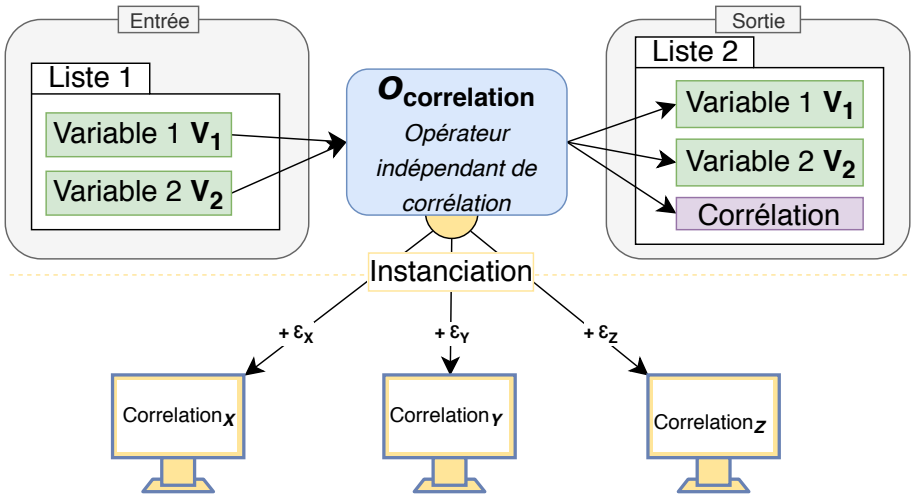


Figure 7.1.: Illustration du fonctionnement d'un opérateur indépendant de corrélation dans notre approche, appliqué sur des variables en entrée. Il produit en sortie une nouvelle variable représentant la corrélation entre V_1 et V_2. Utiliser un opérateur de corrélation implémenté dans un outil d'analyse pour instancier l'opérateur indépendant revient à l'enrichir des spécificités ε propres à cet outil.

thodologie de construction des connaissances visées par l'analyse, indépendamment des contraintes techniques.

Pour mettre en œuvre un tel processus indépendant, il faut réaliser chacune des étapes décrites par les opérateurs indépendants. Cela est possible car chaque opérateur indépendant pointe vers des opérateurs implémentés dans différents outils d'analyse. La mise en œuvre d'un processus d'analyse indépendant peut ainsi utiliser des outils d'analyse différents pour les différentes étapes du processus. Cela offre d'ailleurs plusieurs avantages à l'analyste, comme le fait d'identifier si l'analyse est réalisable dans son ou ses outils de prédilection, ou limiter le nombre d'outils nécessaires à l'analyse. La Figure 7.2 illustre nos propos concernant cette instanciation et les processus d'analyse indépendants.

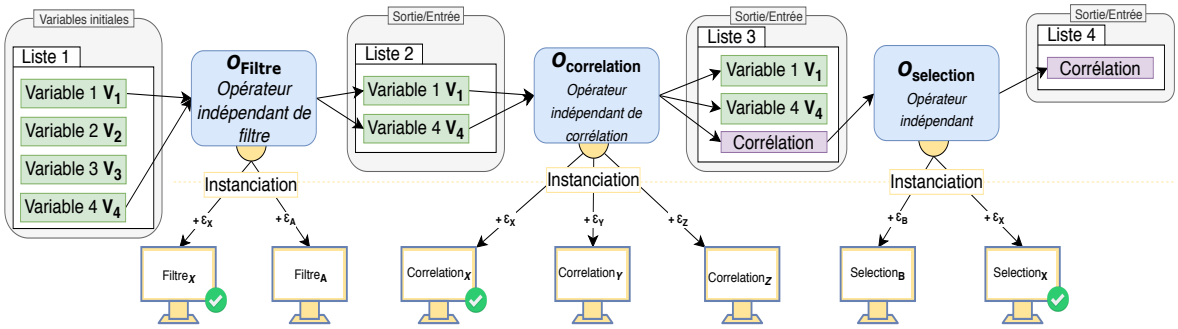


Figure 7.2.: Illustration de l'enchaînement d'opérateurs indépendants constituant un processus d'analyse indépendant. Ici, le processus peut être mis en œuvre entièrement dans l'outil d'analyse **X** (cf. coches vertes), et partiellement dans les autres outils **A**, **B**, **Y** et **Z**.

7.2 Méthodologie de création des méta-modèles

Les méta-modèles que nous proposons dans la section suivante sont issus d'une étude empirique à la fois des divers outils d'analyse et des opérateurs qu'ils implémentent. En effet, il est nécessaire, pour construire de tels méta-modèles, d'avoir identifié les caractéristiques communes aux concepts véhiculés par les opérateurs, l'impact des outils sur les opérateurs et la manière générale dont une analyse est menée, puisqu'il existe plusieurs méthodes pour la mettre en œuvre (*e.g.* workflow, langage de programmation). Pour identifier ces différentes caractéristiques, nous avons adopté une démarche

exploratoire et incrémentale au sein des outils d'analyse Orange : Data Mining, UnderTracks, R, RapidMiner et kTBS [2].

Nous avons tout d'abord étudié le contexte d'utilisation de ces outils d'analyse, afin d'identifier d'éventuelles spécificités quant aux besoins d'analyse auxquels ils permettaient de répondre. Définir ces spécificités en fonction des outils induit en effet un critère discriminant dans l'établissement d'un modèle d'opération indépendante, puisque des particularités techniques peuvent motiver l'utilisation d'un outil – et donc des opérations qu'il implémente.

Suite à cela, nous avons procédé à un inventaire exhaustif, autant que faire se peut, des opérateurs dans ces différents outils d'analyse, dans l'objectif d'opérer une mise en correspondance entre les opérateurs qui sont similaires (*i.e.* ceux partageant un même concept d'opération). Cet inventaire s'est d'abord constitué d'un recensement des opérateurs présents puis de l'identification de leurs caractéristiques principales (*e.g.* objectif, nom, description), pour être en mesure de les regrouper correctement.

Puis, nous avons étudié l'implémentation et le fonctionnement de ces opérateurs au sein de leurs outils respectifs. Nous nous sommes particulièrement intéressé à comment un opérateur définissait le ou les types de données nécessaires à son fonctionnement, les prenait en entrée, à la manière dont il était paramétré, ainsi que les données qu'il produisait en sortie. De plus, étudier les erreurs et la manière dont elles étaient gérées nous a permis d'affiner le cadre applicatif de ces opérateurs.

Lors de cet inventaire, nous avons défini une fiche expérimentale pour regrouper et formaliser par opérateur les différentes caractéristiques regardées que nous avons évoquées jusque-là. Une telle fiche est présentée en Annexe A, Section A.1. En outre, nous avons étudié comment les données étaient représentées dans les outils d'analyse, ainsi que leur format. Conjointement prises en compte, ces fiches et études des données nous ont permis de formaliser les concepts d'opérations rencontrés dans les différents outils pour, *in fine*, élaborer les méta-modèles présentés ci-après. La fiche d'identification du concept d'opération présentée en Annexe A, Section A.2, illustre cette formalisation.

7.3 Formalisation du méta-modèle CAPTEN-ALLELE

Dans cette section, nous décrivons CAPTEN-ALLELE (met**A** mode**L**s for technica**L**ly ind**E**pendent ana**L**ysis proc**E**sses) : l'ensemble de nos méta-modèles qui, lorsque utilisés conjointement, nous permettent de décrire un processus d'analyse indépendamment des outils d'analyse et de leurs aspects techniques. Nous présentons également la manière de procéder pour interpréter l'effet d'un opérateur sur des variables dans le cadre abstrait de notre approche. Cela nous permet de rappeler que nous ne calculons pas directement les données : nous laissons cette tâche complexe, sujette de maints travaux, aux outils d'analyse.

7.3.1 Méta-modèle d'une variable

L'analyse de traces d'apprentissage, l'on en convient, concerne directement des données, afin d'en extraire de potentielles connaissances destinées à être réutilisées. Bien qu'abstraite, il est alors impératif que notre approche permette de maintenir cette notion de données dans l'analyse, ainsi que de représenter leurs évolutions au cours de l'analyse, au risque autrement de ne pas pouvoir la définir correctement. Comme nous l'avons présenté ci-avant, nous ne cherchons pas à importer directement les traces puisque cela impliquerait alors soit l'import des contraintes techniques liées à ces dernières, soit la définition d'un formalisme commun de traces – qui est hors du travail de cette thèse tant cette question revêt une incroyable complexité (J. C. Stamper et al., 2011 ; Advanced Distributed Learning, 2013 ; Dimitrakopoulou, 2004).

Nous avons donc choisi d'éliciter les concepts contenus dans les traces – les variables, au détriment des valeurs qu'elles contiennent. Chaque concept est ainsi représenté dans notre approche conjointement

2. Nous n'avons étudié que théoriquement Usage Tracking Language car l'outil d'analyse associé n'était pas implémenté lors de notre étude.

par un qualificatif, usuellement le nom, qui permet de les symboliser auprès de l'analyste, et par son type (*e.g.* un entier, une chaîne de caractères). Ce couple est enrichi par un identifiant unique – utilisé principalement pour désambigüiser le concept lors de conflits (*e.g.* deux concepts semblables), et par un identifiant de conteneur – nous expliquons ce concept dans la sous-section suivante. La Figure 7.3 présente notre méta-modèle des variables.

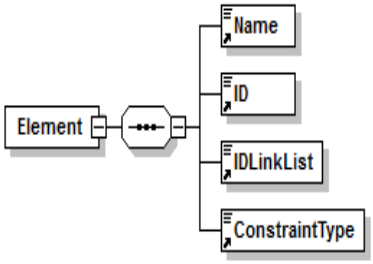


Figure 7.3.: Vue schématique du Document Type Definition (DTD) définissant le méta-modèle de variable.

Concrètement, dans une trace qui contient des dates de connexions d'étudiants à un dispositif pédagogique, comme 05-01-1941 ou 14/06/18, nous pouvons définir une variable nommée *DateConnexion* qui conceptualise ces différentes valeurs. Un tel élément de la trace serait alors du type *xs :date*, une date.

7.3.2 Méta-modèle d'une liste

La conséquence d'éliciter les concepts contenus dans les traces est qu'ils deviennent pour ainsi dire indépendants des traces desquelles ils sont issus. Or, la trace en elle-même constitue un élément important de l'analyse, que ce soit sur le plan structurel (*e.g.* une trace par étudiant) que sur le plan sémantique (*e.g.* une trace dédiée aux échanges textuels des apprenants lors d'une activité précise) : elle contient de l'information, l'organise et lui adjoint une sémantique.

Il nous a semblé pertinent de conserver cette caractéristique, tout en préservant les variables de leur indépendance technique. Pour cela, nous avons proposé la notion de liste (*List*), qui est présentée dans la Figure 7.4. Une liste est donc un conteneur de variables non contraignant : elle constitue le seul lien structurel qui existe entre les variables ; une variable n'étant associée qu'à une seule liste (cf. *IDLinkList*). En outre, une liste fait office de support sémantique libre induit par son nom, laissée à l'interprétation de l'acteur (*e.g.* l'analyste).

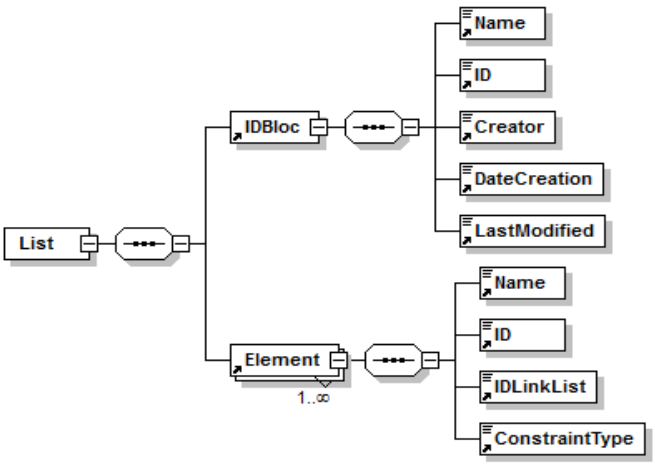


Figure 7.4.: Vue schématique du Document Type Definition (DTD), définissant le méta-modèle de liste.

En plus de cela, une liste possède des métadonnées supplémentaires pour favoriser sa traçabilité et le partage entre les différents acteurs : son identifiant, son créateur (*e.g.* l'analyste, le système), sa date de création et de dernière modification. Enfin, une liste est composée d'un ensemble non ordonné non vide de variables. Par exemple, l'on pourrait utiliser une liste pour regrouper toutes les informations d'une trace relatives aux apprenants, et une autre au dispositif pédagogique.

7.3.3 Méta-modèle d'un opérateur indépendant

Nous l'avons vu, un opérateur indépendant est l'élicitation du concept d'opération qui est commun entre différents opérateurs qui existent dans des outils d'analyse quelconques : dit simplement, il s'agit alors du dénominateur commun de ces opérateurs similaires. La Figure 7.1 a été donnée en guise d'illustration.

Néanmoins, un opérateur n'en reste pas moins un artefact informatique complexe, qui matérialise son concept d'opération par une séquence d'instructions prédéfinies mises en œuvre dans un outil d'analyse. Ces particularités techniques s'ingèrent, comme cité précédemment, jusque dans la configuration même des opérateurs (*e.g.* le nombre de variables nécessaires). Or, par nature, un opérateur indépendant joue le rôle d'un modèle générique pour un type précis d'opérations : l'opérateur implémenté n'en devient alors qu'une spécialisation dans un outil donné.

Il est donc nécessaire de pouvoir exprimer plus que la sémantique d'un concept d'opération pour représenter convenablement un opérateur implémenté. Ce dernier possède des prérequis fondamentaux à son utilisation qui doivent être retranscrits, ainsi que sa validité d'utilisation. De plus, l'effet escompté de son utilisation sur les traces doit aussi être représentable. Alors, l'on dote l'opérateur indépendant de suffisamment d'attributs pour qu'il puisse être utilisé convenablement par les différents acteurs lors de la description de l'analyse dans un cadre abstrait.

Pour cela, notre étude des outils d'analyse et de leurs opérateurs (cf. Section 7.2) nous a permis d'identifier les caractéristiques communes nécessaires à ces opérateurs indépendants. Nous en proposons le méta-modèle d'opérateur indépendant illustré dans la Figure 7.5.

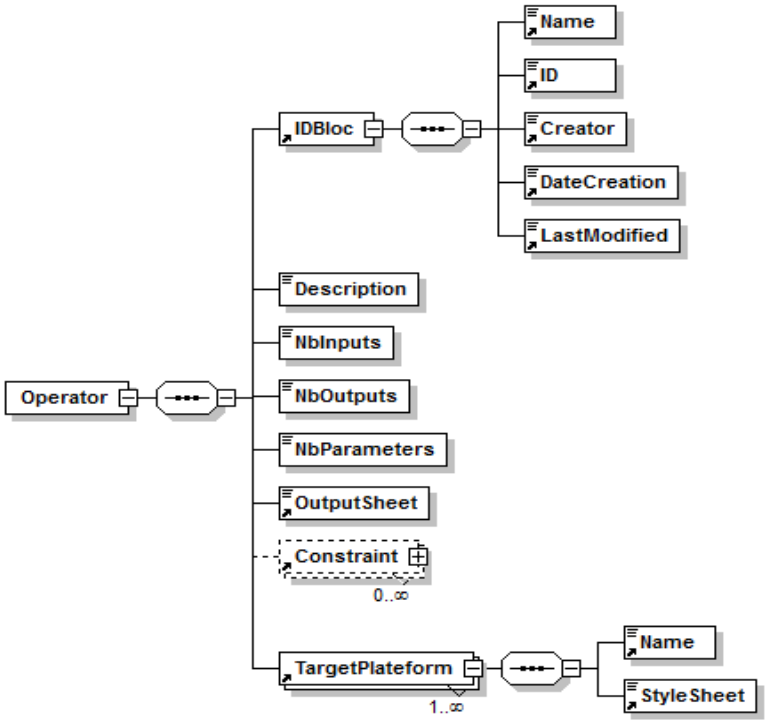


Figure 7.5.: Vue schématique du Document Type Definition (DTD) définissant le méta-modèle de l'opérateur indépendant.

Ce méta-modèle permet donc de définir les prérequis d'utilisation d'un opérateur indépendant. Il les définit en fonction du nombre de variables qu'un opérateur indépendant accepte en entrée (*NbInputs*), et du nombre de paramètres supplémentaires (*NbParameters*) nécessaire à son utilisation. En outre, ce méta-modèle permet l'expression de contraintes (*Constraint*) additionnelles sur la qualité des variables qui seront passées en entrée, ou des paramètres. Il permet aussi d'expliciter la sémantique du concept d'opération représenté par l'opérateur *via* un nom et sa description, sous forme de texte libre.

Nous permettons également de représenter l'impact qu'auront les opérateurs indépendants lorsqu'ils sont utilisés sur les variables, et ainsi représenter l'évolution des variables au cours de l'analyse. Pour cela, nous attribuons à chaque opérateur un archétype comportemental (*OutputSheet*) qui est fonction de différents éléments (*e.g.* variables d'entrée). De cet archétype comportemental résultent deux conséquences : (1) l'indication du nombre de variables de sortie (*NbOutputs*) par rapport au nombre de variables d'entrée et de paramètres attendus dans la définition de l'opérateur indépendant et (2) la production de ces variables lorsque l'opérateur indépendant est utilisé dans un processus d'analyse indépendant. Nous reviendrons sur cette notion d'archétype comportemental, sur la manière de le construire et sur les règles de production des variables en Section 7.3.5.

Par ailleurs, nous exploitons la propriété des opérateurs indépendants de tenir lieu de modèle générique pour un type d'opération en pointant vers les outils qui les implémentent dans leur contexte technique – et qui sont donc directement exécutables. Dans notre méta-modèle, cela se traduit par l'attribut *TargetPlatform*. Il permet d'indiquer le ou les outils d'analyse qui supportent un opérateur indépendant *via* son nom, et le ou les opérateurs qui l'implémente(nt) *via* un patron explicatif[3] (*StyleSheet*).

Pointer vers les outils d'analyse qui implémentent les opérateurs indépendants crée des conséquences intéressantes. Cela offre des perspectives quant à la création d'un inventaire des opérateurs disponibles au sein de la communauté qui, en plus d'être parcourable en fonction des outils d'analyse, peut aussi l'être en fonction des concepts d'opérations affranchis des contraintes techniques. Mais cela permet aussi et surtout à l'analyste de connaitre les outils capables de réaliser une opération spécifique, et de choisir celui qu'il préfère : à l'échelle d'un processus d'analyse, cette propriété est importante, puisqu'elle permet d'assurer une certaine consistance dans le choix des outils.

7.3.4 Méta-modèle d'un processus d'analyse indépendant

Un processus d'analyse de traces d'apprentissage se compose d'une séquence ordonnée d'opérateurs appliqués sur des données. Étant alors un artefact composite, il souffre d'autant de contraintes techniques que d'éléments qu'il fait intervenir pour le définir et qui sont implémentés dans des outils spécifiques. Dès lors, nous proposons le méta-modèle illustré en Figure 7.6 qui permet de définir le concept de processus d'analyse indépendant : un artefact composite exploitant les concepts de variables indépendantes, de listes et d'opérateurs indépendants pour s'affranchir des contraintes techniques qui pésent sur leurs homologues implémentés.

Un processus d'analyse indépendant est essentiellement composé d'une séquence d'opérateurs qualifiés de "configurés". Entendez pour "opérateur configuré" qu'un opérateur indépendant (*Operator*) est mis en relation avec des variables (*Input*) et éventuellement des paramètres (*Parameter*). En outre, en accord avec l'archétype comportemental de cet opérateur indépendant, les variables de sortie (*Outputs*) sont explicitement présentes – et donc exploitables par les opérateurs indépendants suivants.

Un processus d'analyse indépendant a vocation à être défini *a posteriori* d'une analyse mise en œuvre dans un outil : il représente donc toute la chaîne de traitement à opérer sur des variables, et la manière dont elles évoluent pour mener à l'obtention des variables finales de l'analyse – potentiellement les connaissances recherchées. Cet enchaînement d'opérateurs nous permet donc de matérialiser la relation de dépendance qui existe entre les variables finales et celles initiales – disponibles dans les traces. Au final, cela permet de modéliser les variables qui sont nécessaires pour réaliser, dans un outil d'analyse, une analyse décrite par un processus d'analyse indépendant.

3. L'objectif final de ce patron est de fournir les éléments nécessaires pour une instanciation automatique de l'opérateur indépendant vers l'outil d'analyse et son opérateur relatif. Néanmoins, dans les contraintes temporelles de la thèse, nous nous sommes limités à des informations descriptives permettant une instanciation manuelle des opérateurs indépendants.

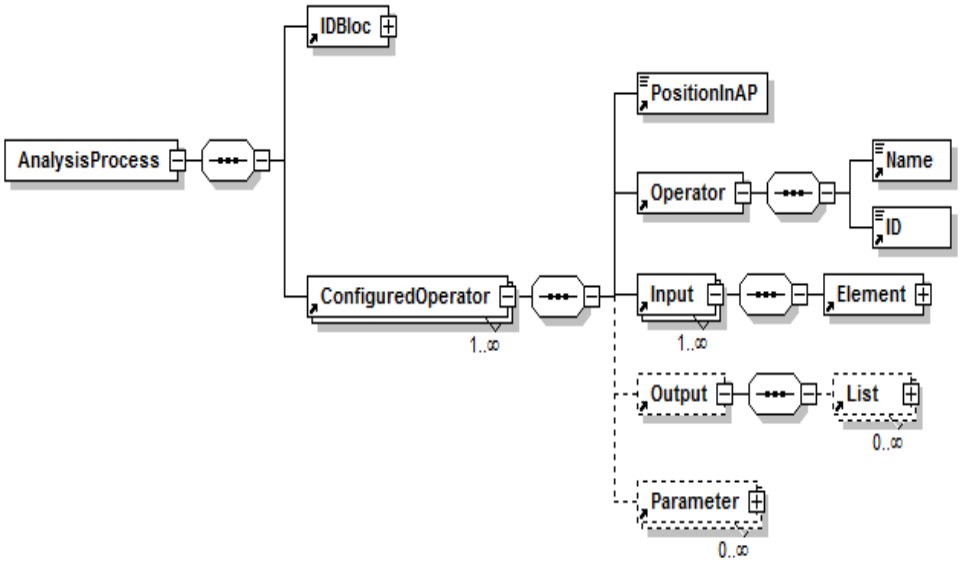


Figure 7.6.: Vue schématique du Document Type Definition (DTD) définissant le méta-modèle des processus d'analyse indépendant.

Enfin, nous fournissons en Annexe B une vue aussi exhaustive que possible de nos méta-modèles et de leurs interactions pour définir un processus d'analyse indépendant.

7.3.5 Fonctionnement et grammaire

L'archétype comportemental d'un opérateur indépendant est défini par une OutputSheet, comme présenté dans la Section 7.3.3. Elle sert à décrire comment les variables qui seront passées en paramètre vont évoluer. Plus spécifiquement, une OutputSheet décrit les variables nouvelles suite à l'application d'un opérateur, celles qui sont conservées par rapport à celles en entrée, et celles que l'opérateur va supprimer – et dans tous les cas les nouvelles listes associées.

Puisqu'il s'agît d'un archétype, l'*OutputSheet* n'est pas instancié sur des variables précises. À l'inverse, nous utilisons un système de token pour identifier les variables (*i.e.* $\$e_x$) en fonction de leur position lorsqu'elles sont passées en paramètre pour former un opérateur configuré. Pour s'assurer que la bonne variable est utilisée sur le bon "port" d'entrée de l'opérateur indépendant, le champ *Constraint* du méta-modèle d'opérateur indépendant est capital, puisqu'il permet de fournir les informations adéquates. De même, nous identifions les listes auxquelles les variables appartiennent par le même système de token (*i.e.* $\$e_x.list$, puisqu'une variable n'est associée qu'à une seule liste).

Les OutputSheet sont strictement définis par la grammaire présentée en Grammaire 7.1. Nous avons défini trois instructions de base qui, lorsque combinées, nous permettent de représenter les comportements de tous les opérateurs que nous avons été amenés à implémenter – nous reviendrons sur l'implémentation dans la Partie III de ce manuscrit. Ces trois instructions sont : (1) créer (*CREATE*) une nouvelle variable, (2) conserver (*MAINTAIN*) la variable d'entrée sans la modifier et (3) supprimer (*DELETE*) la variable d'entrée.

Lorsqu'un opérateur indépendant est appliqué sur des variables en entrée (*i.e.* lorsqu'un opérateur configuré est décrit), l'OutputSheet est alors consommée suivant l'Algorithme 1 présenté. Cela a pour effet de produire les variables de sortie d'après les variables fournies en entrée, et de les placer dans les nouvelles listes qui vont contenir les variables disponibles – plusieurs listes peuvent être produites si les variables fournies proviennent de listes différentes.

Algorithm 1: Production des variables de sortie d'un opérateur indépendant

Input: O_c : Opérateur indépendant configuré
Output: Nouvelles listes d'éléments indépendants $L_{new} = \langle l_1 \dots l_n \rangle$

Function faireEvoluer(O_c)
  $sheet \leftarrow getOutputSheet(O_c)$
  $nbLists \leftarrow numberOfOutputs(sheet)$
  $L_{new}\,[nbLists]$
  for $i \leftarrow 0$ ***to*** $nbLists$ **do**
    `/* Récupère les instructions de création de la liste d'éléments` $L_{new}[i]$
      `présentes dans la feuille de style (cf. Grammaire 7.1), puis la crée. */`
    $instructions \leftarrow parseSheetFor(sheet, i)$
    $L_{new}[i] \leftarrow genererListe(instructions)$
  return L_{new}

Input: $\mathcal{I}$: Instructions de création de la liste
Output: Nouvelle Liste d'éléments indépendants l

Function genererListe($\mathcal{I}$)
  $l \leftarrow createEmptyList()$
  for $i \leftarrow 0$ ***to*** $|\mathcal{I}|$ **do**
    $e \leftarrow extractTokenElement(instructions[i])$
    $liste \leftarrow extractTokenList(instructions[i])$
    $type \leftarrow getInstructionType(instructions[i])$
    if *match(type, "MAINTAIN")* **then**
      if $liste \neq \emptyset$ **then**
        $l \leftarrow copy(liste)$
      else
        $l \leftarrow e$
    else if *match(type, "CREATE")* **then**
      $l \leftarrow e$
    else if *match(type, "DELETE")* **then**
      if $liste \neq \emptyset$ **then**
        for $j \leftarrow 0$ ***to*** $liste|$ **do**
          if $liste[j] \neq e$ **then**
            $l \leftarrow liste[j]$
  return l

⟨start⟩ ⊨ CREATE ⟨tok_integer⟩ List. ⟨lists⟩ !
⟨lists⟩ ⊨ LIST$⟨tok_integer⟩ : ⟨instructions⟩ |
LIST$⟨tok_integer⟩ : ⟨instructions⟩ ⟨lists⟩
⟨instructions⟩ ⊨ ⟨instruction⟩. | ⟨instruction⟩. ⟨instructions⟩
⟨instruction⟩ ⊨ MAINTAIN {$⟨tok_entity⟩, $⟨tok_list⟩} |
DELETE {$⟨tok_entity⟩} |
CREATE {$⟨tok_entity⟩, (⟨tok_name⟩, ⟨tok_type⟩)}
⟨tok_integer⟩ ⊨ ⟨decimal⟩ | ⟨tok_integer⟩ ⟨decimal⟩
⟨tok_name⟩ ⊨ ⟨characters⟩
⟨tok_type⟩ ⊨ ⟨characters⟩
⟨tok_entity⟩ ⊨ e⟨tok_integer⟩
⟨tok_list⟩ ⊨ e⟨tok_integer⟩.list
⟨characters⟩ ⊨ ⟨character⟩ | ⟨characters⟩⟨character⟩
⟨character⟩ ⊨ A | ... | Z
⟨decimal⟩ ⊨ 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

Grammaire 7.1.: Gammaire, dans la forme de Backus-Naur, du langage utilisé dans les OutputSheets pour définir le comportement des opérateurs indépendants.

7.4 Illustration

Dans cette section, nous illustrons la mise en œuvre de notre proposition. Considérons comme exemple le besoin d'analyse suivant : "*Obtenir un indicateur relatant le pourcentage de visionnage d'une vidéo par une promotion d'étudiants sur un intervalle de temps, du 08/10 au 15/10*" ; considérons également le fait que cette analyse puisse exister et avoir été mise en œuvre dans un outil d'analyse quelconque.

Tout d'abord, la personne en charge de décrire dans notre cadre abstrait l'analyse ainsi réalisée doit identifier, d'après les traces qu'elle possède, les concepts de données (*i.e.* variables) qui sont nécessaires pour conduire l'analyse. Elle en extrait par exemple les variables *Temps* et *VisionnageEffectué*. Cela revient à éliciter les concepts contenus dans les traces en variables : elles constitueront alors l'ensemble des variables initiales nécessaires au processus d'analyse indépendant. Nous pensons que cette tâche incombe principalement à l'expert de l'EIAH qui a connaissance des traces, ou à l'analyste (cf. Figure 6.4, page 90).

Ensuite, cette personne en charge doit définir le processus d'analyse indépendant, c'est-à-dire quels opérateurs indépendants sont utilisés, et comment (*e.g.* sur quelles variables les appliquer). Cette étape est complexe et ardue, puisqu'il est nécessaire d'opérer une mise en relation des opérateurs implémentés avec les opérateurs indépendants, pour savoir ceux qu'il convient d'utiliser. Du fait de sa propension à requérir une expertise d'analyse plus importante, il convient d'imaginer l'analyste comme acteur principal de cette étape. Ici, la possibilité de connaître les outils d'analyse et les opérateurs associés pour chaque opérateur indépendant (cf. *TargetPlatform* dans la Figure 7.5) se révèle précieuse lors de l'élicitation.

Cette mise en correspondance avec les concepts d'opération a pour effet de mettre en valeur les étapes de l'analyse [4]. En reprenant le besoin énoncé, les étapes seraient alors :

1. Isoler la semaine recherchée ;
2. Filtrer les apprenants ayant vu la vidéo ;
3. Compter les apprenants ayant vu la vidéo ;
4. Compter le nombre total d'apprenants ;

4. Cette notion d'étapes sera introduite et formalisée dans le Chapitre 8 suivant.

5. Diviser le compte d'apprenants ayant vu la vidéo par le compte total d'apprenants.

Chaque élément de cette liste fait intervenir un opérateur indépendant appliqué sur une ou plusieurs variables. Pour compter les apprenants ayant vu une vidéo, l'on appliquera alors sur la variable conceptualisant le visionnage un opérateur indépendant destiné à compter indépendamment les occurrences. Un tel opérateur produit en sortie un nombre d'occurrences qui, dans notre cas, représentera le nombre d'étudiants ayant vu une vidéo précise.

Cette étape est visible dans la partie gauche de la Figure 7.7. Il s'agit d'une capture d'écran de notre prototype lors de la description de ce processus d'analyse. Nous distinguons ici qu'un opérateur indépendant *Compter Distinct* est appliqué sur une variable *vidVisio*. Grâce à l'archétype comportemental de l'opérateur indépendant, le système sait qu'une nouvelle variable *occurs* est produite – il est possible de la renommer pour préciser sa sémantique. Nous reviendrons plus en détail sur ce prototype dans la Partie III au sujet de la mise en œuvre de nos contributions.

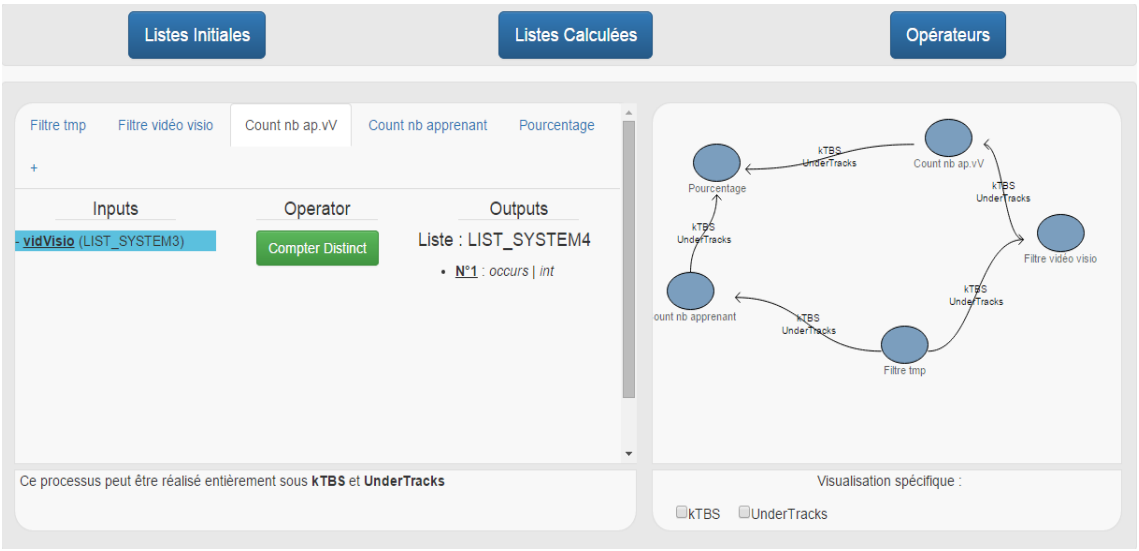


Figure 7.7.: Capture d'écran de notre prototype implémentant notre proposition théorique à propos de l'indépendance technique des processus. Le processus d'analyse indépendant du calcul du pourcentage de visionnage d'une vidéo y est décrit. La partie gauche représente l'application d'un opérateur indépendant sur des variables. La partie droite est une représentation graphique du processus d'analyse, où les outils d'analyse implémentant les opérateurs indépendants sont indiqués.

Ainsi, on obtient un processus d'analyse indépendant qui décrit la manière d'obtenir la connaissance attendue, indépendamment des contraintes techniques. Ce processus, du fait de l'abstraction qui est opérée tout au long de sa chaîne de traitement, sert de modélisation générique à l'obtention de cette connaissance. Il devient donc pertinent de partager un tel processus au sein de la communauté. Par exemple, il pourra aider les décideurs dans leur étape de formalisation du besoin d'analyse. Pour s'en servir concrètement, il convient alors de l'instancier dans un ou plusieurs outils d'analyse qui supportent la totalité des opérations (dans la Figure 7.7, les outils kTBS et UnderTracks).

Ce qu'il faut retenir

Dans ce chapitre, nous avons présenté l'approche **CAPTEN-MANTA** et sa partie théorique, **CAPTEN-ALLELE**, permettant de :

— s'affranchir des contraintes techniques ;

— couvrir les propriétés de réplicabilité et de reproductibilité nécessaire à la capitalisation.

Cette approche s'appuie sur quatre méta-modèles :

— celui de **variable**, permettant de décrire les concepts présents dans les traces ;

— celui de **liste**, permettant d'organiser les variables et de leur attacher une sémantique commune ;

— celui d'**opérateur indépendant**, permettant de décrire le concept d'opération commun entre des opérateurs similaires mais implémentés dans des outils d'analyse différents ;

— celui de **processus d'analyse indépendant**, faisant intervenir les trois concepts précédents, et qui permet de décrire une analyse indépendamment des contraintes techniques induites par les outils d'analyse.

8 Capitaliser les processus d'analyse de traces : de leur abstraction à leur narration

Sommaire

Section	**Introduction**	**108**
Section 8.1	**Description de l'approche**	**109**
8.1.1	Représentation structurelle	110
8.1.2	Représentation contextuelle	112
Section 8.2	**Méthodologie de construction de l'ontologie**	**114**
Section 8.3	**Notre framework ontologique CAPTEN-ONION pour la narration**	**115**
8.3.1	Réplicabilité	115
8.3.2	Répétabilité	118
8.3.3	Compréhension	118
8.3.4	Réutilisation	119
8.3.5	Ouverture	121
8.3.6	Adaptabilité	122
Section 8.4	**Illustration**	**124**
8.4.1	Narration des traces	124
8.4.2	Narration des étapes de l'analyse	125
8.4.3	Narration de l'analyse	126
Section	**Ce qu'il faut retenir**	**127**

Publications relatives à ce chapitre

(Lebis et al., 2017a) A. Lebis, M. Lefevre, V. Luengo et N. Guin (2017a). "Approche narrative des processus d'analyses de traces d'apprentissage : un framework ontologique pour la capitalisation". In : *Environnements Informatiques pour l'Apprentissage Humain*. EIAH 2017. Strasbourg, France

(Lebis et al., 2018a) A. Lebis, M. Lefevre, V. Luengo et N. Guin (2018a). "Capitalisation of Analysis Processes : Enabling Reproducibility, Openess and Adaptability thanks to Narration". In : *LAK '18 - 8th International Conference on Learning Analytics and Knowledge*. Sydney, Australia : ACM, p. 245–254

Introduction

Dans le Chapitre 7, nous avons présenté notre première contribution visant à affranchir les processus d'analyse de traces des caractéristiques techniques qui les contraignent. Les résultats expérimentaux (cf. Partie IV, Chapitre 13) produits suite à l'utilisation de notre prototype (cf. Partie III, Chapitre 10) par les analystes sont encourageants et semblent fortement confirmer la nécessité d'adopter un nouveau paradigme pour pouvoir capitaliser les processus d'analyse. Il est possible de tirer de ces résultats deux conclusions majeures, qui ont servi de matrice à la proposition présentée dans le reste de ce chapitre.

La première conclusion est qu'adopter une démarche d'abstraction des processus d'analyse pour favoriser leur indépendance technique sied à la capitalisation, puisqu'elle satisfait les propriétés de réplicabilité et de répétabilité.

La deuxième conclusion est que cette première proposition manque d'informations et de sémantique. Cela a entravé la compréhension par les analystes des différents éléments de l'analyse, et a limité les possibilités de réutilisation des processus d'analyse indépendants : puisque la sémantique des concepts n'y est pas définie de manière univoque, l'apparition d'une divergence sémantique entre l'idée de l'analyste et processus d'analyse indépendant ou entre processus d'analyse implémenté et indépendant a été fréquemment observée.

Aussi, ces conclusions nous permettent de dire que les contraintes techniques ne constituent pas la seule limitation à la capitalisation des processus d'analyse (Chatti et al., 2012). Il ne suffit donc pas de résoudre le problème de dépendance technique d'un processus d'analyse pour le qualifier de capitalisable. En effet, la situation pédagogique sur laquelle a été conduite l'analyse est intégrée de manière implicite aux processus d'analyse implémentés. Cela révèle à la fois des besoins de compréhension des différents aspects de l'analyse (*e.g.* objectif, contexte d'apprentissage, configurations d'opérations) et des besoins de représenter ces informations de manière intelligible à la fois pour l'humain et la machine.

Or, présentement, les outils d'analyse existants ne répondent pas aux besoins de compréhension des acteurs de l'analyse lors de la réutilisation des processus d'analyse, mais uniquement au besoin computationnel que l'utilisation occasionne. Effectivement, l'on a observé (cf. Chapitre 2) que ces outils d'analyse ne permettent pas toujours de décrire l'information, ou alors soit seulement certaines informations – prédéfinies lors du développement de l'outil – soit d'une manière non structurée (*e.g.* commentaires).

En outre, ces outils sont principalement destinés aux analystes. Or, comme présenté dans la Section 6.1.1, l'analyse des traces fait intervenir et collaborer différents types d'acteurs, chacun avec son expertise. La prise en compte des choix et des besoins de ces acteurs dans la description du processus d'analyse devient pour ainsi dire indissociable d'une représentation intelligible. Néanmoins, ces informations sont complexes par nature, comme les choix réalisés par l'expert de l'environnement pédagogique pour fournir les meilleures traces à l'analyste. Elles peuvent aussi être interdépendantes, comme l'adoption d'une politique d'anonymisation du décideur qui nécessite une étape de prétraitement sur les traces.

Ce manque de compréhension a un impact négatif direct sur tous les acteurs de l'analyse lors de la réutilisation des processus, comme nous l'avons déjà exposé dans la Section 6.1.3. Il est donc important de pouvoir matérialiser ces informations, et de les organiser entre elles (*e.g.* montrer qu'un choix dépend d'une spécificité du contexte pédagogique). Il l'est tout autant d'adopter une sémantique compréhensible par l'ensemble des acteurs ; compréhensible également pour le système, afin de lui permettre d'assister convenablement ces acteurs en raisonnant avec tous les éléments constitutifs de l'analyse.

Notons toutefois que les travaux présentés dans l'état de l'art (cf. Section 4.1 page 42) apportent des pistes intéressantes pour renforcer la compréhension et tenir compte de ces différentes informations. C'est par exemple le cas des Research Object (Page et al., 2012) qui permettent d'associer aux workflows des ressources supplémentaires, structurées par une ontologie. C'est aussi le cas de

l'ontologie SWAN (Ciccarese et al., 2008), qui permet d'exprimer sémantiquement des informations scientifiques, comme des hypothèses ou encore des inconsistances. Néanmoins, ni ces ontologies, ni leur granularité ne permettent de représenter les éléments du contexte qu'il serait nécessaire de décrire pour la réutilisation des processus d'analyse de traces d'apprentissage.

Dans ce chapitre, nous présentons notre seconde proposition qui vise à obtenir une capitalisation effective des processus d'analyse de traces d'apprentissage. Pour cela, nous nous sommes inspirés des travaux susmentionnés pour enrichir notre première approche destinée à proposer une indépendance technique des processus d'analyse. Nous définissons ainsi un nouveau paradigme pour représenter les processus d'analyse : la *narration* des processus d'analyse. Il s'agit d'en modéliser les différentes informations, survenant à différents moments du cycle d'élaboration de l'analyse, de manière structurée grâce à une ontologie – ce qui revient à raconter l'histoire de la mise en œuvre de l'analyse dans des codes prédéfinis.

Cette ontologie nous octroie la possibilité de représenter le contexte pédagogique et les informations de l'analyse d'une manière sémantisée. Par exemple, l'on peut apporter des éléments décrivant les travaux scientifiques justifiant l'utilisation d'un opérateur d'analyse donné pour mettre en œuvre une étape d'un processus d'analyse. Cela nous permet aussi de représenter les processus d'analyse eux-mêmes de manière sémantisée, et ainsi décrire leurs différents éléments constitutifs, les relations qui existent entre eux, mais aussi entre toutes les informations supplémentaires. Enfin, adopter une ontologie nous offre des possibilités de raisonnement accrues pour la machine, comme nous le verrons dans le chapitre suivant.

Aussi, dans la Section 8.1, nous décrivons notre approche narrative supportée par une ontologie. Nous détaillons dans la Section 8.2 la méthodologie suivie pour établir cette ontologie. Dans la Section 8.3, nous formalisons notre proposition adossée aux propriétés de la capitalisation, et aussi la manière dont l'ontologie est utilisée pour décrire les analyses. Enfin, nous illustrons cette proposition dans la Section 8.4.

La mise en œuvre de cette proposition et les résultats expérimentaux associés sont présentés respectivement dans le Chapitre 11, page 153, et dans le Chapitre 14, page 179.

8.1 Description de l'approche

Nous définissons la narration d'un processus d'analyse comme la représentation sémantique de ses éléments structurels et contextuels. Il s'agit respectivement de pouvoir modéliser les aspects techniques des différentes étapes de l'analyse, à l'instar de notre approche précédente, et de modéliser les différentes informations supplémentaires et pertinentes, par exemple celles issues des choix effectués par les acteurs durant l'élaboration de l'analyse ou celles décrivant le contexte de la situation pédagogique.

Cette narration porte sur le processus d'analyse, mais aussi sur chacun de ses éléments constitutifs (*e.g.* étapes, opérateurs, paramètres). Elle concerne également les relations qui existent entre ces éléments, ainsi que dans les traces analysées. De cette manière, nous visons une représentation des processus qui permet l'intégration de l'information et sa mise en relation directe avec l'analyse, afin de créer un artefact auto-suffisant. Il en résulte qu'un processus d'analyse ainsi narré devient compréhensible intrinsèquement, favorisant aussi sa réutilisation et son adaptation, puisque les spécificités de configuration sont explicitées.

Pour rendre cette narration possible, nous avons défini une ontologie des processus d'analyse de traces d'apprentissage qui structure à la fois les processus d'analyse et les informations associées à ces processus. Nous avons choisi une modélisation ontologique parce que les ontologies [1] ont une propension naturelle à modéliser précisément un domaine de connaissances au travers d'éléments

1. Une ontologie, d'un point de vue informatique, permet la modélisation formelle d'un système (*i.e.* de ses entités pertinentes et de ses relations observables) *via* des classes et des propriétés. Il s'agit alors de la spécification formelle d'une conceptualisation partagée. Une ontologie possède un ensemble d'axiomes terminologiques (*i.e.* TBox) qui définit les caractéristiques du système modélisé et des règles d'assertions pour spécifier les caractéristiques des individus (Staab et Studer, 2010)

sémantiques définis de manière univoque, et à permettre des possibilités de raisonnement et d'inférence complexes. De plus, les travaux comme wf4ever (BELHAJJAME et al., 2018) ou OPMW (GAJIRO et GIL, 2010), qui reposent sur des ontologies, ont montré de bons résultats, et, d'après notre état de l'art, sont les plus proches de ce que nous considérons comme la capitalisation des processus d'analyse (cf. Section 5 page 65). Par conséquent, notre ontologie a été définie en réutilisant des termes et des relations provenant d'autres travaux, comme wf4ever, prov, SWAN ou encore xAPI ; l'objectif était d'agir dans le cadre d'une interopérabilité forte, pour promouvoir la propriété d'ouverture des processus d'analyse.

8.1.1 Représentation structurelle

Dans notre proposition précédente concernant l'indépendance technique des processus d'analyse (cf. Chapitre 7), nous nous sommes concentrés sur trois éléments structurels majeurs de l'analyse, à savoir (1) les traces, (2) les opérateurs et (3) le processus d'analyse, sans tenir compte de la sémantique qu'ils pouvaient véhiculer. Or, nous l'avons observé (cf. Chapitre 3), cela empêche la capitalisation des processus d'analyse. Il est donc nécessaire de pouvoir expliciter la sémantique qui réside dans ces éléments et aussi dans leur utilisation. Par exemple, l'utilisation d'un opérateur sur des variables forme une étape dans l'analyse – or l'on se doute qu'une étape possède un objectif propre qui motive l'utilisation de l'opérateur qui la forme et que cet opérateur est configuré d'une certaine manière en fonction des variables.

L'objectif de la narration est de pouvoir expliciter ces informations. Pour ce faire, nous avons construit une ontologie qui conserve les particularités de notre première approche, tout en l'enrichissant avec de nouveaux éléments. Il en résulte que l'entièreté de notre première proposition peut être projetée dans notre ontologie, ce qui signifie qu'un processus d'analyse indépendant peut être assimilé à un sous-ensemble d'un processus d'analyse modélisé dans notre ontologie – un processus d'analyse narré.

Pour faire émerger la sémantique contenue dans les traces, nous avons conservé le concept de **variable** comme étant une représentation abstraite d'un élément d'une trace. Nous y ajoutons néanmoins une dimension sémantique en utilisant des éléments de vocabulaire directement issus de notre ontologie pour qualifier ces variables. En outre, ces éléments de vocabulaires peuvent provenir d'autres ontologies, comme celle d'xAPI.

De plus, pour enrichir la description de cette trace, nous avons décidé de remplacer la notion de liste introduite par notre première approche en mettant les variables d'une trace directement en relation entre elles. De cette manière, il est par exemple possible de représenter la sémantique de la donnée "Étudiant de MOOC" d'une trace comme étant une variable "Étudiant" liée à la variable "MOOC" par la relation "impliquée dans". Le choix des relations exprimant la sémantique des éléments de la trace est effectué en consultant les traces elles-mêmes, mais également la situation pédagogique dont elles sont issues.

Nous obtenons ainsi un **graphe de variables** : un ensemble de variables sémantisées représentant les éléments de la trace, et leurs interconnexions – explicites ou implicites dans la trace. Un exemple de graphe de variables est visible dans la partie (a) de la Figure 8.1. Ici, ce graphe exprime le fait qu'un "Étudiant de MOOC", d'un certain "Age", interagit avec un "Cours de MOOC" et que ce MOOC décerne une certification (*Grade*). On remarque également que l'étudiant peut regarder des vidéos dans le cours de MOOC, et que ses connexions journalières, issues d'une agrégation préalable des traces, sont comptées.

De même, pour faire émerger la sémantique contenue dans les opérateurs, nous avons enrichi le concept d'opérateur indépendant en **opérateur narré** et revu la manière dont un opérateur s'exprime dans l'analyse. Il s'agit toujours de représenter un concept d'opération commun partagé par des opérateurs similaires, mais cette fois-ci en l'identifiant avec un qualificatif sémantiquement défini dans l'ontologie. Conscient néanmoins que l'archétype comportemental d'un opérateur – son impact attendu – intervient invariablement dans la définition même de la sémantique du concept d'opération, nous en

proposons une nouvelle représentation afin qu'il puisse s'exprimer, tout en conservant la possibilité de représenter l'impact des opérations sur les variables.

Aussi, un opérateur narré possède un patron d'entrée, un patron de sortie et un patron de configuration. Le patron d'entrée d'un opérateur narré représente les variables attendues en entrée ; les variables sont définies par des graphes de variables et exploitent le vocabulaire de l'ontologie – et donc les relations sémantiques qui peuvent exister entre les variables. Cela permet ainsi de décrire le cadre d'application dans lequel il est théoriquement possible d'utiliser l'opérateur. Le patron de sortie exprime le sens du résultat de l'opérateur narré, là aussi *via* des graphes de variables. De plus, il est également possible de représenter les relations que peuvent partager ces variables de sortie avec celles d'entrées. Le patron de sortie décrit donc l'effet de l'opérateur. Enfin, le patron de paramétrage exprime les configurations de l'opérateur, et permet aussi de représenter l'impact qu'auront ces paramètres sur les variables.

Dans le cas où le résultat d'un opérateur narré est censé produire un modèle en sortie, il est également exprimé à l'aide de concepts sous la forme de graphes de variables et les paramètres nécessaires utilisés sont sauvegardés. En accord avec l'approche de la découverte avec les modèles proposés par Baker *et al.* (R. S. J. D. Baker et Yacef, 2009), l'ensemble des étapes (cf. ci-après) qui conduisent à ce modèle peuvent être réutilisées comme un nouvel opérateur narré. Ce nouvel opérateur définit ainsi une sémantique particulière, qui s'exprime notamment par des configurations et des patrons particuliers.

Un exemple d'opérateur narré est visible dans la partie (b) de la Figure 8.1. Il s'agit d'un opérateur de corrélation, qui, d'après son patron d'entrée, doit être appliqué sur des "Entité Numérique" pour produire un "Coefficient de Corrélation". Ce coefficient est directement influencé par le type de mesure utilisée lors de cette corrélation (*e.g.* coefficient de Pearson).

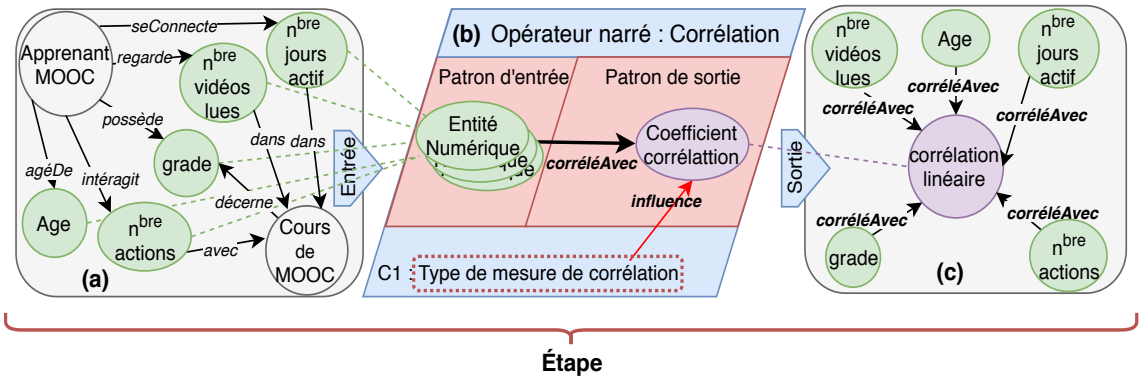


Figure 8.1.: Étape d'un processus d'analyse narré, faisant intervenir un graphe de variables en entrée (a) et un opérateur narré de corrélation (b). La partie (c) est le résultat effectif de l'application de l'opérateur narré sur le graphe de variables (a).

De surcroît, nous proposons dans cette approche de capturer la noèse opérée par l'analyste au niveau atomique de l'analyse, soit l'intention qui a conduit l'analyste à l'utilisation d'un opérateur à un moment précis. Pour cela nous avons explicité et formalisé le concept d'**étape** d'un processus d'analyse : elle représente le traitement que l'on souhaite effectuer, alors mis en œuvre par un opérateur. Une étape est ainsi assimilable à un conteneur qui encapsule les éléments contribuant à la transition d'un état des données manipulées par l'analyste vers un autre état.

Concrètement, une étape est constituée d'un opérateur narré et du graphe de variables sur lequel cet opérateur est appliqué. En résulte alors un graphe de variables (cf. (c) dans la Figure 8.1), dit de sortie, qui représente les variables qui sont potentiellement disponibles après l'application d'un opérateur narré ; donc comment les variables en entrée ont évolué après application de l'opérateur narré. Les modifications effectuées sur le graphe de variables d'entrée sont directement imputables aux patrons de l'opérateur utilisé. Les nouvelles variables d'un graphe de variables de sortie, ou celles modifiées, sont elles aussi sémantiquement définies par des termes issus du vocabulaire de l'ontologie.

Toutefois, les transformations automatiques inférables par les patrons ne sauraient dispenser les acteurs de l'analyse d'une intervention manuelle sur la définition des variables obtenues dans une

étape. Les acteurs peuvent eux aussi modifier le graphe de sortie pour affiner la sémantique des variables produites. Par exemple, le patron de sortie d'un opérateur narré "*Division réelle*" définit l'obtention d'une nouvelle variable sémantiquement définie comme étant le *Quotient*. Or, dans un contexte où le dividende d'une division est la somme des notes d'un apprenant et le diviseur le nombre de notes, il ne s'agit alors plus d'un simple *Quotient*, mais d'une *Moyenne* qui spécialise *Quotient*, et plus particulièrement d'une *Moyenne* concernant des notes qu'il convient d'indiquer dans le graphe de variables de sortie. Nous laissons donc la possibilité aux acteurs de dériver la manière dont s'expriment les opérateurs narrés en fonction des contextes. Nous reviendrons sur ce point en Section 8.3.6.

Ainsi, l'ensemble de la Figure 8.1 représente une étape d'un processus d'analyse narré. Un opérateur narré de corrélation est utilisé sur des variables du graphe d'entrée, qui représente l'état des variables à un instant précis de l'analyse, pour produire en sortie une information précise : la corrélation linéaire entre ces éléments.

Enfin, un **processus d'analyse narré** représente l'ensemble des traitements qu'il est nécessaire d'effectuer sur les traces pour produire les connaissances souhaitées. Dans notre approche, cela revient à définir un processus d'analyse narré comme une succession ordonnée d'étapes, et non plus simplement d'opérateurs indépendants. De plus, nous y ajoutons la possibilité d'identifier sémantiquement les variables qui tiennent le rôle de connaissances attendues (*i.e.* le résultat de l'analyse). Nous pouvons ainsi définir intuitivement le cadre de validité d'un processus narré (*i.e.* variables d'entrées nécessaires à l'analyse) par le patron d'entrée de la première étape du processus, et les connaissances attendues par le patron de sortie de la dernière étape.

8.1.2 Représentation contextuelle

La première partie que l'on vient de voir permet de définir un ensemble structurel cohérent et sémantisé pour intégrer les informations supplémentaires nécessaires à la capitalisation des processus d'analyse. Pour renforcer la propriété d'ouverture amorcée par cet ensemble, et pour satisfaire à la propriété de compréhension, et ainsi mener vers une adaptabilité des processus, nous y ajoutons des éléments narratifs.

Comme nous l'avons vu dans l'état de l'art, les informations relatives à l'analyse élaborée (*e.g.* contexte pédagogique, choix de conceptions) sont décrites de manière spécifique à chaque outil. De plus, ces informations sont le plus souvent données avec une faible granularité et en annexe du processus d'analyse, par exemple sous la forme d'un commentaire textuel libre qui ne concerne explicitement aucun élément du processus : cela occasionne une information qui ne couvre pas correctement l'analyse. C'est d'ailleurs ce qui est pointé par Belhajjame & *al*. (Belhajjame et al., 2012b) comme l'une des causes de non-réutilisation de l'existant : adopter une approche sémantique et structurée pour décrire l'information fournit une meilleure compréhension et réutilisation des analyses.

Notre hypothèse est que ces informations doivent effectivement être structurées, mais aussi décrites de manière non ambigüe et intégrées directement au processus d'analyse – et non pas considérées comme une ressource annexe. Ainsi il est possible de décrire les informations relatives au contexte dans lequel l'analyse a été conduite et les incidences de ce contexte sur les choix de conception effectués lors de l'élaboration de l'analyse ; l'on obtient alors un processus d'analyse auto-suffisant.

Nous proposons donc que notre ontologie ait un pan dédié à la représentation des informations de l'analyse. Pour permettre cela, nous introduisons les trois concepts principaux que sont : (1) les **éléments narratifs**, (2) le **vocabulaire contrôlé** et (3) les **propriétés sémantiques**. La Figure 8.2 présente comment ces trois concepts cohabitent pour intégrer dans un processus d'analyse des informations contextuelles, et ainsi le qualifier de narré. Cette figure reprend l'exemple du processus d'analyse destiné à prédire la certification dans un MOOC, présenté dans la Section 6.1.2, page 83.

Un élément narratif est défini comme le support de description d'une information spécifique : il représente un type d'information et ce qui en est dit. Par exemple, sur la Figure 8.2, l'on note un élément narratif du type *Objectif*, sa sémantique étant de représenter l'objectif de l'élément auquel il

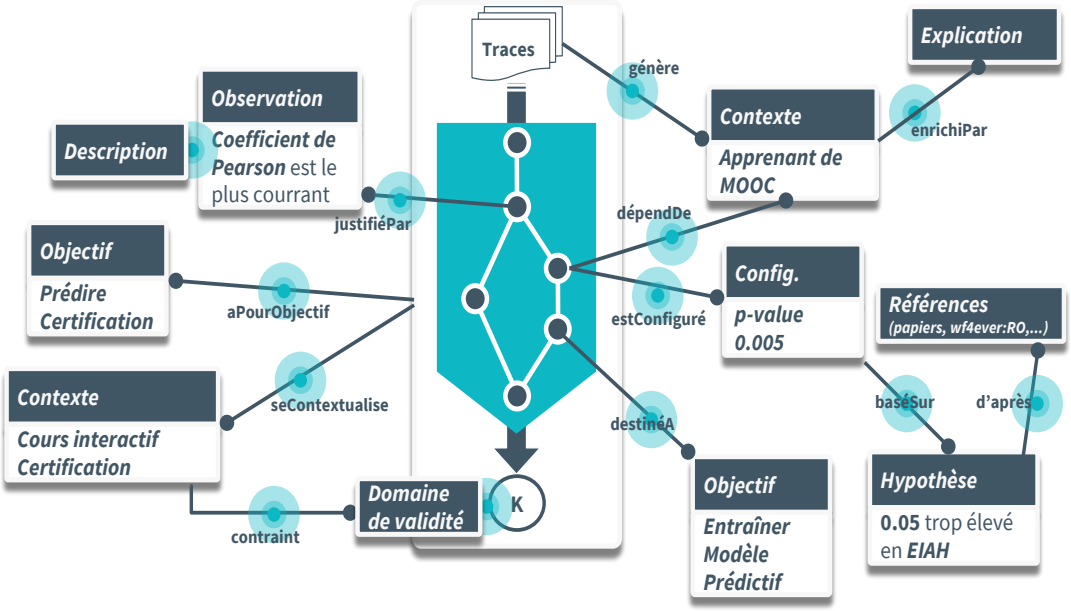


Figure 8.2.: Exemple d'une narration décrivant un processus d'analyse visant à prédire la certification des étudiants dans un MOOC. Les éléments narratifs sont représentés par les boîtes d'information. L'encart d'une boîte (ici en foncé) représente le type de l'élément narratif, et le contenu de la boîte (en clair) est la description narrative, où les éléments en gras correspondent à des éléments du vocabulaire. Ces éléments narratifs sont associés au processus d'analyse, à l'une de ses étapes (nœuds en noir), ou à d'autres éléments narratifs.

est associé – ici le processus d'analyse narré [2]. L'explicitation de l'élément narratif (*i.e.* sa "valeur") est définie à partir d'un ensemble de termes provenant d'un vocabulaire contrôlé intégré à notre ontologie, éventuellement complété par du texte libre. L'utilisation d'un tel vocabulaire permet d'obtenir des informations interprétables par la machine, et non ambigües pour l'humain. Dans la Figure 8.2, l'élément narratif précédent *Objectif* est décrit avec deux termes du vocabulaire contrôlé, à savoir *Prédire* et *Certification*.

L'ontologie que nous avons définie prévoit ainsi un ensemble d'éléments narratifs que nous avons prédéfinis, afin d'enrichir la description d'un processus d'analyse. Toutefois, cet ensemble est susceptible d'être étendu selon les besoins de la description. Les nouveaux types d'éléments narratifs ainsi ajoutés peuvent également être mis en relation avec ceux existant dans l'ontologie pour enrichir les possibilités descriptives.

Notre ontologie fait intervenir la notion de vocabulaire contrôlé pour décrire l'ensemble de l'analyse, comme les variables ou le contenu narratif. Le vocabulaire contrôlé se définit comme un ensemble de termes (*e.g.* individus, classes, au sens ontologique du terme) et de propriétés sémantiques entre ces termes (*i.e.* relations), utilisables pour décrire les divers éléments de l'analyse, majoritairement dans le cadre des Learning Analytics. Chaque élément est ainsi sémantiquement défini, ce qui contribue à réduire l'ambigüité des descriptions et à fournir un support stable pour le raisonnement machine (*e.g.* favoriser le partage grâce à une interopérabilité des termes). Bien que le vocabulaire soit prédéfini par nos soins, il est destiné à s'étoffer au fur et à mesure des besoins de la communauté ; *in fine* il en émergera potentiellement un inventaire et un consensus sur le vocabulaire appartenant au domaine.

La définition des éléments du vocabulaire et leur utilisation s'illustrent par exemple dans la Figure 8.2. Considérons l'élément narratif *Contexte* associé à *Traces*, en haut à droite, spécifiant ici le contexte des traces analysées (un autre élément contextuel est attaché au processus lui-même, en bas à gauche).

2. Il est tout à fait possible dans notre ontologie d'associer un objectif à une étape, à un opérateur narré, à une configuration, *etc*.

Il précise que ces traces sont issues de l'activité de *Apprenant de MOOC*. Ce terme est défini dans le vocabulaire à la fois par *isA(Apprenant de MOOC, Apprenant)* – signifiant qu'un apprenant de MOOC est un apprenant, et par *impliqué(Apprenant, MOOC)* – signifiant qu'un apprenant est impliqué dans un MOOC. Comparativement à du texte libre, une telle définition de *Apprenant de MOOC* nous permet ainsi d'inférer une particularité supplémentaire à l'analyse, à savoir qu'elle concerne les *MOOC*.

De surcroît, nous permettons de modéliser les propriétés sémantiques qui existent entre un élément narratif (*e.g. Objectif*) et n'importe quel autre élément constitutif du processus d'analyse : graphe de variables, opérateur, étape, processus d'analyse, et même un autre élément narratif. Concrètement, nous relions ainsi sémantiquement deux éléments à l'aide d'une relation définie dans notre vocabulaire contrôlé. Chaque propriété est donc labélisée par un terme qui spécifie la relation qui existe entre l'élément narratif et l'élément qu'il décrit. Par conséquent, il devient possible d'organiser sémantiquement l'information au sein du processus d'analyse narré.

La Figure 8.2 fournit un exemple des possibilités qu'offre cette approche. D'une manière générale, l'on y distingue divers éléments narratifs associés sémantiquement à diverses entités de l'analyse. Toutefois, nous y présentons aussi comment des informations complexes sont structurées dans notre approche, comme avec le cas de la configuration. L'élément narratif *Configuration* est associé à un opérateur d'une étape du processus d'analyse narré, et est aussi mis en relation avec un élément narratif *Hypothèse* – lui aussi relié à des références. Concrètement, cet ensemble permet de traduire que la configuration qui a été adoptée dépend d'une hypothèse qui a été formulée d'après un recueil de références scientifiques.

Ainsi, les éléments narratifs, organisés *via* des propriétés sémantiques et s'appuyant sur un vocabulaire partagé, permettent de décrire de manière structurée chaque élément du processus d'analyse et les raisons qui ont conduit à l'élaboration de ce processus. De plus, elle offre par nature différents niveaux de lecture, s'adressant aux différents acteurs de l'analyse. Il devient en effet possible d'expliquer les choix effectués lors des différentes étapes de l'analyse, ou encore son contexte. Cette approche narrative, réifiée par notre ontologie, nous permet de satisfaire les propriétés nécessaires à la capitalisation (Lebis et al., 2018[b]).

8.2 Méthodologie de construction de l'ontologie

L'ontologie que nous proposons dans la Section 8.3 pour permettre la capitalisation des processus d'analyse *via* leur narration a été élaborée en exploitant diverses sources. Elle est issue d'une part de nos travaux précédents et d'une étude empirique de différentes propositions sémantiques, comme ^{My}Experiment, afin d'identifier les différentes limites concernant la capitalisation, et les raisons associées. D'autre part, notre approche résulte d'une étude empirique des différents scénarios d'analyse rencontrés dans le cadre du projet HUBBLE (Projet HUBBLE, 2018). Avoir à disposition des cas concrets a eu pour avantage d'apporter un référentiel terrain non négligeable lors de l'élaboration de l'ontologie.

Nous avons tout d'abord élaboré l'ontologie à partir des méta-modèles des processus d'analyse indépendants, afin de conserver leurs propriétés de réplicabilité et de reproductibilité. Pour s'en assurer, nous avons décrit les processus d'analyse indépendants dans cette ontologie ; puis nous avons cherché et étudié les divergences de description qui pouvaient survenir entre processus équivalents décrits dans l'approche indépendante et narrative. Cela a par exemple permis de mettre en exergue le caractère inconstant de la paternité des variables dans l'approche indépendante, imputable notamment au concept de liste.

D'une manière générale, ces comparaisons nous ont servi de référence lors de la conception de l'ontologie. À chacune de ses modifications, nous avons ainsi pu nous assurer de conserver les deux propriétés que nous avions validées, à savoir la réplicabilité et la répétabilité.

Suite à cela, nous avons procédé à un inventaire des différentes manières de structurer l'analyse dans les ontologies des processus d'analyse. Comme nous l'avons vu, des travaux comme P-Plan (Garijo et Gil, 2012) ou OMPW (Gajiro et Gil, 2010) adoptent une représentation de la structure de l'analyse

qui leur est propre, certaines fois interopérables voire équivalentes, d'autres fois non. Il était donc essentiel selon nous d'observer quels étaient les points communs de ces différentes modélisations, qui constituent alors une entente tacite potentielle – quelquefois même pluridisciplinaire, et de les mettre en rapport avec notre ontologie. Les concepts distincts entre ces travaux, témoins de la richesse de modélisation des analyses, ont également été inventoriés et étudiés pour extraire des éléments structurants jugés nécessaires dans le cadre des Learning Analytics.

Puis, nous avons étudié les possibilités descriptives qu'offraient ces différentes modélisations quant aux informations supplémentaires. Nonobstant les Research Objects et les informations de publication, nous avons pu constater un réel manque d'informations liées aux processus d'analyse et également un manque dans la possibilité de les organiser. Aussi, pour établir notre approche narrative, nous avons élargi le spectre des ontologies considérées.

Ces propositions étant très disparates, conserver une démarche de comparaison n'était pas adapté. À la place, nous avons cherché des critères discriminants permettant de statuer sur l'utilisation de certaines modélisations plutôt que d'autres, comme la complexité des ontologies, la couverture descriptive des analyses ou encore la modélisation de l'analyse (*e.g.* sous forme de workflow, de plan). Puis, nous avons utilisé les scénarios rencontrés dans HUBBLE pour identifier les éléments informatifs qui nous semblaient pertinents, ainsi que la manière de les organiser. Par exemple, les travaux de Ciccarese & *al.* (Ciccarese et al., 2008) et de Freire & *al.* (Freire et al., 2006) concernant la qualité scientifique des workflows ont servi de base dans la définition des éléments narratifs scientifiques, comme les hypothèses, avant que nous les enrichissions avec des éléments sémantiques supplémentaires, nécessaires dans notre domaine (*e.g.* objectif pédagogique de l'analyse).

8.3 Notre framework ontologique CAPTEN-ONION pour la narration

Dans cette section, nous présentons notre framework ontologique pour capitaliser les processus d'analyse de traces d'apprentissage par la narration et comment, de manière formelle, la narration est façonnée. Pour appuyer cette présentation, nous l'opérons de manière incrémentale vis-à-vis des différentes propriétés de la capitalisation (cf Figure 6.2 de la Section 6.2, page 88). L'objectif est ainsi de faciliter l'identification des éléments capitaux de l'ontologie pour chacune de ces propriétés.

Afin de servir de support à notre formalisation, nous proposons dans la Figure 8.3 une transposition simplifiée de notre ontologie dans le format UML. La version complète de notre ontologie est accessible en ligne (Lebis et al., 2018[b]). Notons que cette ontologie réexploite les travaux de *PROV*, *xAPI* ou *wf4ever*, toujours dans l'optique de favoriser une meilleure interopérabilité entre les différents travaux et technologies et *in fine* entre les différentes communautés. De ce seul fait, cette ontologie n'est pas à considérer comme définitive : elle est destinée à se raffiner grâce à l'aspect communautaire, intrinsèque, de la capitalisation ; nous n'en proposons ici que le point de départ.

8.3.1 Réplicabilité

D'après la définition de la réplicabilité que nous avons donnée dans la Section 6.2, page 87, un processus est réplicable s'il est possible d'y décrire tous ses opérateurs et l'ordre qui existe entre eux. Les deux éléments principaux de notre ontologie qui œuvrent pour cette propriété de réplicabilité sont en toute logique l'opérateur narré et le processus d'analyse narré.

Ainsi, à l'instar d'un opérateur indépendant, et puisqu'il en découle, un opérateur narré $\mathcal{No}$ élicite un concept commun entre différents opérateurs qui existent dans différents outils d'analyse. Toutefois, pour favoriser la structuration et l'intégration de l'information (cf. Section 8.3.3 et 8.3.5), nous avons défini un concept de haut niveau représentant les éléments de notre framework $\mathcal{F}$: l'opérateur narré est subsumé par ce concept. En outre, nous apportons aussi une sémantique à ce concept élicité, afin de le rendre non ambigu et de pouvoir ensuite l'exploiter. Par exemple, un opérateur narré de

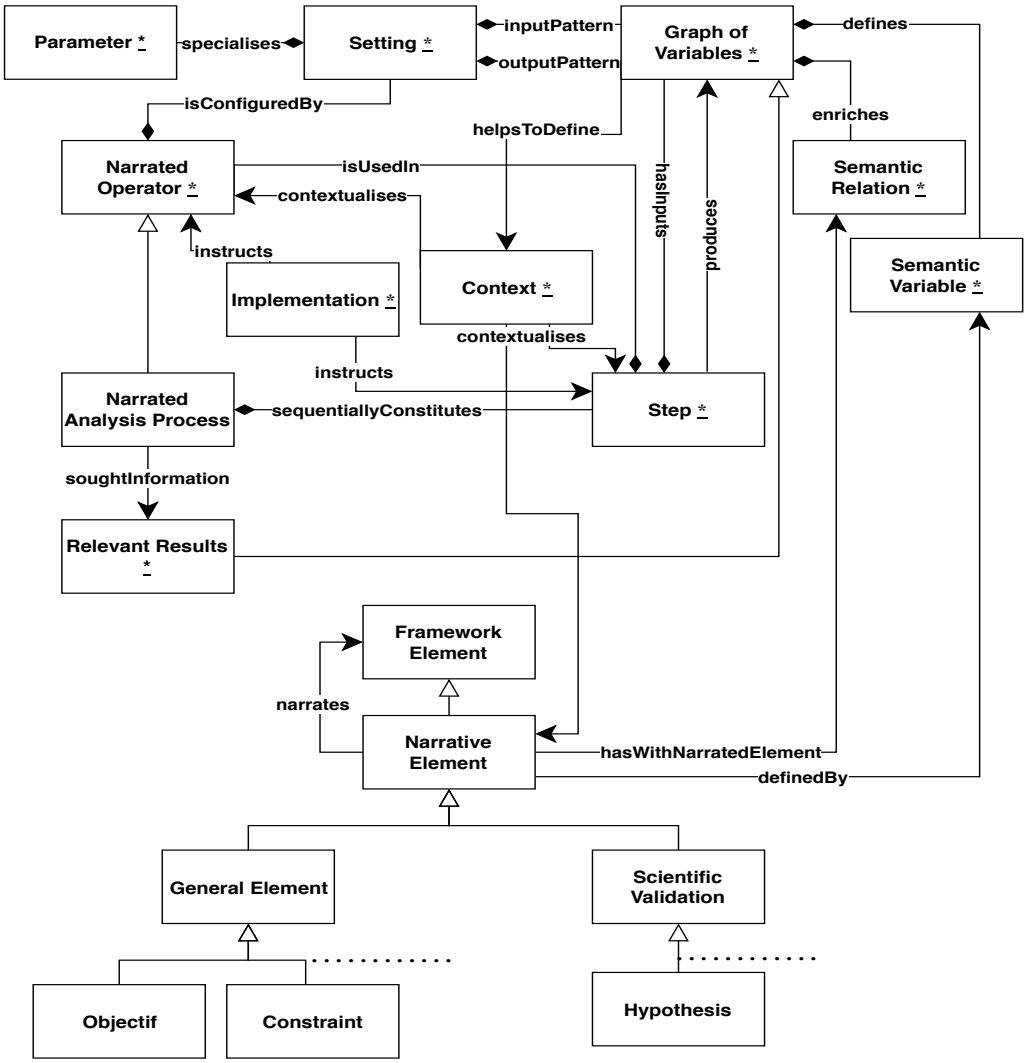


Figure 8.3.: Transposition simplifiée de notre ontologie dans le format UML. Le symbole * signifie *hérite de Framework Element*.

corrélation pourra être défini en utilisant des termes sémantiques provenant de l'ontologie STATO (STATO, 2018[a]), bénéficiant ainsi des propriétés qui leur sont associées.

De plus, un opérateur narré s'exprime différemment dans cette approche, comme illustré dans la Figure 8.3. En effet, puisqu'un opérateur est complexe par nature, il est nécessaire de pouvoir représenter correctement les prérequis fondamentaux à son utilisation. Or, notre expérimentation précédente nous a montré que ces prérequis sont complexes et, si mal modélisés, peuvent entraîner un fossé sémantique important.

Pour résoudre ce problème, nous définissons un opérateur narré comme étant configuré par un patron d'entrée $\mathcal{P}e$, un patron de sortie $\mathcal{P}o$ et un patron de paramétrage $\mathcal{P}p$. Grâce à ces patrons, nous pouvons respectivement expliciter le cadre d'application de l'opérateur, l'impact estimé de l'opérateur narré et le paramétrage et ses effets. Ces patrons sont définis à l'aide de graphes de variables (cf. Section 8.3.2), ce qui nous permet d'indiquer la sémantique des variables attendues et de potentiellement utiliser les relations sémantiques des variables à disposition dans une analyse pour les faire correspondre aux patrons.

Brièvement aussi, car nous reviendrons sur ces éléments en détail dans les propriétés auxquelles ils contribuent le plus, un opérateur narré est associé à des concepts d'implémentation $\mathcal{I}m$, à un contexte d'utilisation $\mathcal{C}u$ et, par transitivité par l'entremise des étapes, à un processus d'analyse. Avec le concept

d'implémentation, nous conservons la possibilité d'indiquer les outils d'analyse qui implémentent un opérateur narré, cette fois-ci avec la possibilité d'avoir des relations supplémentaires, comme une utilisation particulière d'un opérateur narré dans une version précise d'un outil d'analyse. Avec le contexte d'utilisation, nous pouvons indiquer le champ applicatif de l'opérateur narré (*e.g.* statistique, *data mining*).

Définition 3.1 (Opérateur Narré)

Une formalisation d'un opérateur narré $\mathcal{No}$ représentant un concept d'opération $\mathcal{C}$ est donnée par le 6-tuple suivant :

$$\mathcal{No} = \langle n, \mathcal{Pe}, \mathcal{Po}, \mathcal{Pp}, \mathcal{Im}, \mathcal{Cu} \rangle \tag{8.1}$$

Avec $n \in \mathcal{W}$ la sémantique de l'opérateur narré (*e.g.* opérateur de corrélation), issue du vocabulaire $\mathcal{W}$.
$\mathcal{Pe}, \mathcal{Po}, \mathcal{Pp}$, respectivement les patrons d'entrée, de sortie et de paramétrage de l'opérateur narré.
$\mathcal{Im}, \mathcal{Cu}$, respectivement les concepts d'implémentation et le contexte d'utilisation.

Enfin, un processus d'analyse narré représente l'ensemble des traitements qu'il est nécessaire d'effectuer sur des variables pour répondre au besoin d'analyse qui le conditionne. Autrement dit, il existe, pour une analyse mise en œuvre dans un outil d'analyse, autant d'opérateurs narrés que nécessaire pour représenter ses traitements. Mais il est important de conserver en mémoire qu'un processus d'analyse narré représente également les informations *en lien* avec ces traitements.

Il découle de cela qu'un processus d'analyse narré s'applique sur un ensemble de variables initiales $\mathcal{Vi}$, et en produit de nouvelles *in fine* : les connaissances $\mathcal{K}$. Chaque traitement est représenté par une étape spécifique du processus d'analyse narré. Nous sauvegardons la séquence de ces étapes entre elles à l'aide d'une relation sémantique dédiée [3], et créons ainsi un graphe de dépendance de ces étapes. Nous conservons donc l'aspect composite de l'analyse, comme évoqué dans le chapitre précédent, ainsi que son indépendance technique.

Définition 3.2 (Processus d'analyse narré)

Soit $\mathcal{Na}$ un processus d'analyse narré, $\mathcal{E}$ l'ensemble des étapes du processus d'analyse narré, tel que $\mathcal{No} \in \mathcal{E}$. On peut alors définir la chaîne de traitements $\mathcal{Op}$ de $\mathcal{Na}$ tel que :

$$\mathcal{Op} = \langle \mathcal{E}, \mathcal{R}, \gamma', < \rangle \tag{8.2}$$

Avec $\mathcal{R} \in \mathcal{W}$ la relation entre deux étapes, issue du vocabulaire. $\gamma' : \mathcal{R} \to \mathcal{E} \times \mathcal{E}$ qui associe à chaque relation une paire d'étapes $\mathcal{E}$, et $<$ une relation d'ordre partiel sur les étapes de l'analyse.

Nous définissons un processus d'analyse narré comme :

$$\mathcal{Na} = \langle \mathcal{Gi}, \mathcal{Op}, \mathcal{K}, \mathcal{Ne} \rangle \tag{8.3}$$

Avec $\mathcal{Gi}$, le graphe de variables initial, $\mathcal{Op}$ la chaîne de traitements, $\mathcal{K}$ les connaissances produites et $\mathcal{Ne}$ les éléments narratifs associés [a].

Aussi, un processus d'analyse narré n'est pas égal à l'ensemble des opérateurs narrés utilisés pour le définir.

$$\mathcal{Na} \neq \bigcup_i \mathcal{No}_i) \tag{8.4}$$

a. Les graphes de variables et les éléments narratifs sont introduits dans les sections suivantes

Pour permettre d'intégrer de l'information à un processus, et de la structurer, il est nécessaire de pouvoir détailler ses composants, ainsi que de pouvoir les décrire. Pour cela, et c'est là l'une des

3. Nous utilisons la classe sémantique *rdf :Seq* pour représenter cette séquence (W3C, 2014b)

particularités de notre approche, nous subsumons le concept de processus d'analyse narré par celui d'opérateur narré (cf. Figure 8.3), contribuant à des apports significatifs. D'une part, il est possible désormais de couvrir entièrement l'utilisation d'un processus d'analyse comme opérateur d'un autre processus d'analyse ; d'autre part, un processus d'analyse narré possède ainsi un cadre de validité d'application défini par un patron d'entrée, et l'impact qu'il aura – en plus de pouvoir le paramétrer.

8.3.2 Répétabilité

La répétabilité d'un processus d'analyse consiste à observer les mêmes résultats lorsqu'il est de nouveau exécuté. Cela place *de facto* la trace comme un élément certifiant de la répétabilité dans le sens où l'immuabilité des résultats produits par les opérateurs doit être conservée lorsque la même trace initiale est utilisée. Comme nous l'avons vu dans notre précédente proposition, le format des données et leur granularité ont besoin d'être unifiés afin de représenter de manière homogène l'influence des opérations qui y sont appliquées. Cependant, nous avons aussi vu que la sémantique qui existe dans la trace est importante et doit être modélisée.

Le concept de graphe de variables que nous proposons pour modéliser les variables (*i.e.* les concepts de données) de la trace a pour objectif de répondre à ces nécessités. Un tel graphe modélise à la fois la sémantique des variables, et les relations – explicites et implicites – qui existent entre ces variables : il décrit alors conceptuellement la trace, l'affranchissant des contraintes techniques tout en l'enrichissant.

Définition 3.3 (Graphe de Variables)

Un graphe de variables $g \in \mathcal{G}$ est défini comme un graphe orienté, tel que :

$$g = (\mathcal{V}, E) \tag{8.5}$$

où $\mathcal{V}$ représente l'ensemble des variables, et E l'ensemble des arcs $e_n = (v_i, v_j)$ où $v_i, v_j \in \mathcal{V}$.

La définition de la répétabilité (cf. Section 6.2, page 86) fait également intervenir une notion de temporalité dans l'exécution de l'analyse. Dans notre approche, nous représentons le processus d'analyse de manière conceptuelle ; par conséquent, nous ne faisons pas intervenir la notion de temporalité [4]. Si l'on s'interroge, l'on observe que cette temporalité traduit alors le caractère déterministe de l'analyse mise en œuvre, et de toutes ses opérations. Autrement dit, que chaque opérateur produit, avec les mêmes variables fournies en entrée, les mêmes variables qu'il a initialement produites.

Pour traiter ce déterminisme, nous nous reposons sur les graphes de variables pour modéliser ce caractère déterministe au sein de l'analyse que nous représentons. En effet, nous modélisons les patrons $\mathcal{P}e, \mathcal{P}o, \mathcal{P}p$ d'un opérateur narré $\mathcal{N}o$ *via* des graphes de variables (cf. (b) dans la Figure 8.1). Ainsi, de la mise en correspondance d'un graphe de variables avec le patron d'entrée $\mathcal{P}e$ d'un opérateur narré résulte un nouveau graphe de variables $\mathcal{G}o$ qui traduit son impact théorique sur les variables – défini par le patron de sortie $\mathcal{P}o$. Ce comportement est équivalent à une transition d'états du graphe de variables d'entrée $\mathcal{G}e$ de l'analyse, cette transition étant conditionnée par le patron de sortie de l'opérateur narré appliqué : si le même opérateur et appliqué sur les mêmes variables, le résultat sera donc identique.

8.3.3 Compréhension

Les informations relatives aux processus d'analyse implémentés dans les outils d'analyse s'avèrent rares, et le plus souvent non structurées. Nous l'avons vu : c'est le résultat d'un besoin computationnel fort de ces outils (Belhajjame et al., 2012b). Néanmoins, des informations présentes que nous avons pu observer, nous avons remarqué qu'elles s'étendent le plus souvent sur trois dimensions distinctes :

4. La temporalité intervient cependant pour les éléments narratifs, ou pour les différentes versions des éléments du framework (*e.g.* opérateur narré).

une dimension technique, une méthodologique et une en lien avec l'utilisation des résultats produits. Alors que les informations techniques relatent des points spécifiques à l'outil d'analyse et à la mise en œuvre, les deux autres types d'informations apportent une compréhension plus approfondie de l'analyse en elle-même. Celles méthodologiques tendent à expliquer la validité de l'analyse (*e.g.* critère scientifique adopté), et celles concernant l'utilisation sont plus sujettes à prémunir les bénéficiaires d'une mauvaise interprétation des résultats.

Notre approche narrative a pour objectif de représenter ces informations. Comme nous l'avons dit, nous voulons rendre un processus d'analyse narré compréhensible par lui-même. Cela requiert que l'information soit donc directement intégrée à l'analyse – ce qui évite par exemple le risque de la perdre – et qu'elle soit structurée. Pour cela, nous définissons le concept d'élément narratif $\mathcal{N}e$. Un élément narratif représente un type d'information sémantiquement prédéfini (*e.g.* un objectif, une hypothèse) et conditionne ainsi son contenu C. La partie basse de la Figure 8.3 illustre des éléments narratifs.

Définition 3.4 (Élément narratif)

Formellement, un élément narratif $\mathcal{N}e$ se définit comme un couple, tel que :

$$\mathcal{N}e = \langle Type, C \rangle \tag{8.6}$$

Avec $Type \in \mathcal{W}$, le type de l'élément narratif sémantiquement défini, issu du vocabulaire $\mathcal{W}$. C, le contenu de l'élément narratif, constitué d'éléments du vocabulaire $\mathcal{W}$ ou de texte libre.

Malgré l'apparente simplicité des éléments narratifs, ils permettent de décrire des situations complexes au sein d'un processus d'analyse narré. Pour cela, chaque élément narratif doit être relié strictement à un élément de notre framework $\mathcal{F}$. Ce lien, sémantique, définit la relation qui existe entre l'élément narratif et celui narré ; il dépend de l'élément narratif utilisé. De plus, nous définissons le concept même d'élément narratif comme subsumé par celui d'élément du framework : alors il devient possible de narrer un autre élément narratif, et ainsi concevoir des structures narratives complexes. Tout cela s'exprime dans la Figure 8.3 par la relation de subsomption de *Narrative Element* par *Framework Element* et par la présence de la relation *narrates* qui autorise à attacher un élément narratif à n'importe quel élément du framework.

Définition 3.5 (Description narrative des éléments)

La narration d'un élément $x \in \mathcal{F}$ du framework s'exprime comme :

$$e = (\mathcal{N}e, x) \tag{8.7}$$

Avec e l'arc entre l'élément narratif $\mathcal{N}e$ et l'élément narré x.

8.3.4 Réutilisation

Lorsque l'utilisation d'un processus d'analyse n'est plus conditionnée par les traces initiales qui ont servi à son élaboration, alors l'on s'étend au-delà de la propriété de répétabilité. Si les variations contextuelles sont négligeables, nous parlons alors de réutilisation du processus d'analyse (cf. définition de la réutilisation, page 87). Concrètement, cela signifie que l'entièreté de la séquence d'opérations réalisées lors de la mise en œuvre de l'analyse est capable de supporter ces changements et, conséquemment, chaque opérateur : ils sont potentiellement utilisés avec ces nouvelles données.

Dans notre approche, il découle de cela la nécessité de modéliser à la fois l'utilisation de l'opérateur sur les données, mais aussi la noèse qui conduit à cette utilisation, pour pouvoir expliquer et conditionner les configurations de l'opérateur. Nous définissons pour cela la notion d'*étape* de l'analyse. Aussi, une étape $\mathcal{E}$ encapsule un opérateur narré $\mathcal{N}o$ qui est appliqué sur des variables issues d'un graphe de

variables d'entrée $\mathcal{G}e$. Elle traduit également l'impact de l'opérateur narré sur les variables d'entrée en y définissant un graphe de variables de sortie $\mathcal{G}o$. En outre, elle encapsule les différentes informations relatives à l'intention d'analyse qu'elle représente par l'entremise d'éléments narratifs $\mathcal{N}e$.

Définition 3.6 (Étape)

Formellement, une étape $\mathcal{E}$ est définie par le tuple suivant :

$$\mathcal{E} = \langle \mathcal{G}e, \mathcal{N}o, \mathcal{G}o, E_{e,o}, \mathcal{N}e \rangle \tag{8.8}$$

Avec $E_{e,o}$, l'ensemble des arcs entre les variables $v_{(\mathcal{G}e,i)} \in \mathcal{G}e$ mises en correspondance avec les variables $v_{(\mathcal{P}e,i)} \in \mathcal{P}e$ du patron d'entrée de l'opérateur narré $\mathcal{N}o$, tel que $E \in \mathcal{W}$.
$\mathcal{G}o$ est le graphe de variables de sortie produit d'après le patron de sortie $\mathcal{P}o$ de l'opérateur narré $\mathcal{N}o$, et $\mathcal{N}e$ les éléments narratifs qui décrivent l'étape $\mathcal{E}$.

Puisqu'au cours de l'analyse, les variables et leurs données évoluent, il est nécessaire de le traduire dans notre approche. En plus de la fonction d'état-transition γ, nous permettons d'exploiter directement le graphe de variables de sortie d'une étape en tant que graphe de variables d'entrée d'autres étapes. Il en émerge alors une interdépendance des étapes, qui est explicitée grâce au caractère sémantique des relations utilisées [5].

De plus, il devient possible d'opérer une distinction entre les variables nécessaires à l'analyse – qui conditionnent donc le cadre de validité de l'analyse représentée – et les variables d'entrée des étapes. Rigoureusement, ce cadre de validité est défini par l'ensemble des graphes de variables $\mathcal{G}$ qui n'ont pas d'antécédents : ils ne sont donc pas issus de patrons de sortie d'un opérateur narré, ou d'un autre processus.

Définition 3.7 (Connaissance)

Le concept de connaissance $\mathcal{K}$ est subsumé par celui de graphe de variables, tel que :

$$\mathcal{K} \sqsubset \mathcal{G} \sqsubset \mathcal{F} \tag{8.9}$$

Par conséquent, les connaissances $\kappa \in \mathcal{K}$ peuvent être utilisées comme graphe de variables d'entrée dans une étape telle que $\kappa \in \mathcal{G}e$.

Soit $\mathcal{G}o$ l'ensemble des graphes de sortie produits par un processus d'analyse narré $\mathcal{N}a$. Les connaissances κ produites par un processus d'analyse sont définies comme :

$$\kappa \subset \mathcal{G}o \tag{8.10}$$

Tel que les connaissances κ définissent le patron de sortie $\mathcal{P}o$ du processus d'analyse narré.

De ce fait, et des définitions précédentes, il découle qu'un processus d'analyse narré peut aussi être utilisé comme un opérateur narré si et seulement si :

$$\mathcal{N}a = \mathcal{N}o \leftrightarrow \kappa = \emptyset \tag{8.11}$$

Autrement dit, lorsqu'un processus d'analyse narré est utilisé en tant qu'opérateur narré d'un autre processus, les connaissances de ce premier processus d'analyse ne sont pas identifiées comme telles dans l'autre processus – elles ne servent qu'à définir son patron de sortie.

De plus, la propriété d'être réutilisable induit des modifications quant aux connaissances produites par le processus d'analyse. Pour représenter ces modifications, il est nécessaire, dans un premier temps, de pouvoir les modéliser. Pour cela, nous définissons le concept de connaissance K produit par un

5. D'après la Définition 3.2, nous obtenons un graphe orienté d'étapes.

processus d'analyse narré, exprimé aussi d'après un graphe de variables (cf. classe *Relevant Results* dans la Figure 8.3). Ce graphe de variables s'exprime d'après les graphes de variables de sortie des différentes étapes d'un processus d'analyse narré et en forme un sous-ensemble : toutes les variables ou modèles produits par l'analyse peuvent ne pas être pertinents. De plus, il est aussi possible d'utiliser des éléments narratifs pour décrire ces résultats (*e.g.* le contexte de validité attendu des résultats).

Une connaissance est un sous-ensemble des variables disponibles dans les graphes de variables de sortie des différentes étapes du processus d'analyse narré. De plus, il est aussi possible d'utiliser des éléments narratifs pour décrire ces résultats (*e.g.* le contexte de validité des résultats).

8.3.5 Ouverture

Les implémentations disparates des processus d'analyse, induites par les spécificités des outils d'analyse, affectent naturellement leur ouverture, puisqu'il en découle l'impossibilité d'adopter un entrepôt de processus homogènes, dépourvus d'ambiguïtés sémantiques quels que soient les outils d'analyse. Le risque est qu'un processus ne conserve pas sa cohérence scientifique ou technique lorsqu'il est mis à disposition d'autrui, du fait du fossé sémantique des différents termes pour l'entité qui le réutilise (*e.g.* un autre outil d'analyse, un analyste).

L'on peut donc définir l'ouverture par le fait que (1) tous les éléments constitutifs de l'analyse sont déréférencables, (2) qu'il est possible d'identifier sans ambiguïté un même élément de l'analyse indépendamment de l'outil en question et (3) que ces éléments sont immuables dans le temps. Remarquons que, pour ce troisième point, le fait qu'un processus d'analyse soit réplicable contribue partiellement à y répondre.

Pour doter les processus d'analyse narrés de cette propriété d'ouverture, nous avons introduit la notion de vocabulaire contrôlé $\mathcal{W}$. Il s'agit d'un ensemble de termes $w \in \mathcal{W}$ qui sont sémantiquement définis. Pour réaliser cette définition, nous nous reposons sur les propriétés du web sémantique et représentons chacun des termes du vocabulaire contrôlé par une IRI[6]. Par conséquent, il nous est possible de déréférencer les termes utilisés, et aussi d'utiliser des termes issus d'autres travaux, comme xAPI ou wf4ever, pour constituer le vocabulaire contrôlé.

Définition 3.8 (Vocabulaire contrôlé)

Soit $\mathcal{W}$ le vocabulaire contrôlé. Il se définit par :

$$\mathcal{W} = \mathcal{W}c \cup \mathcal{W}p \tag{8.12}$$

Avec $\mathcal{W}c$ et $\mathcal{W}p$ respectivement l'ensemble des classes sémantiques et l'ensemble des propriétés sémantiques.

La conséquence directe est que toutes les relations qui interviennent dans la description d'un processus d'analyse narré s'expriment sous la forme d'une propriété e reliant deux classes (*i.e.* éléments du framework) $v_j, v_k \in \mathcal{F}$ avec un label sémantique w tel que :

$$e_i = (v_j, v_k, w) \tag{8.13}$$

Avec $w \in \mathcal{W}p$.
Chaque élément v est défini par une classe sémantique spécifique tel que $\forall v \exists w_v \in v$, avec $w_v \in \mathcal{W}c$.

Ces termes sémantiques sont utilisés pour décrire les éléments de notre ontologie, dont les différentes relations. Par exemple, les bénéficiaires de l'analyse sont identifiés dans notre ontologie par l'IRI *CAPTEN :TargetUser*. Pour cela, chaque terme du vocabulaire contrôlé est soit une classe sémantique, soit une propriété sémantique. De cette manière, les relations entre les différents éléments de l'ontologie

6. Internationalized Resource Identifier (DÜERST et SUIGNARD, 2005)

sont labélisées ; ce label porte une sémantique, ce qui nous permet d'expliciter la qualité de la relation entre les éléments.

Le vocabulaire contrôlé sert également lors de la description des éléments narratifs. À la place d'utiliser du texte libre pour décrire l'information, nous permettons de la structurer *via* l'utilisation des termes w qu'il est possible de mettre en relation. Un exemple est visible dans la Figure 8.2, page 113. De plus, il est possible d'enrichir le vocabulaire contrôlé avec de nouveaux termes et, *in fine*, de les utiliser dans la définition des processus d'analyse.

8.3.6 Adaptabilité

Finalement, la propriété d'un processus d'analyse à être adaptable constitue le dernier jalon pour le rendre capitalisable. En effet, après s'être assuré de sa réutilisabilité et de son ouverture, il convient de pouvoir le modifier plus fondamentalement, ceci afin de pouvoir l'appliquer à des besoins d'analyse différents – tout en observant une cohérence dans le contexte cible par rapport à celui d'origine. De cela, nous remarquons que la propriété d'adaptabilité est étroitement liée à celle du contexte de l'analyse : les différents éléments de l'analyse sont en effet intriqués aux spécificités des contextes pédagogiques et techniques.

Bien que les éléments narratifs $\mathcal{N}e$ nous permettent d'apporter des informations supplémentaires – mises en relation avec les éléments – il n'en demeure pas moins qu'elles forment indirectement les contextes pédagogiques et techniques : il faut considérer ces informations dans leur ensemble pour espérer les décrire entièrement. Cependant, identifier l'impact d'une situation pédagogique sur l'analyse peut s'avérer complexe. Preuve en est avec les travaux cherchant à attacher des ressources supplémentaires aux workflows (Belhajjame et al., 2015). Pour résoudre cela, nous formalisons directement le contexte $\mathcal{C}$ de l'analyse dans notre ontologie (cf. Figure 8.3), selon trois catégories.

Nous définissons d'abord le contexte pédagogique $\mathcal{C}p$, qui permet d'expliciter la dépendance des éléments de l'analyse vis-à-vis de la situation pédagogique. Nous définissons également le contexte d'utilisation de l'analyse $\mathcal{C}u$ qui permet de définir les situations pédagogiques dans lesquelles une telle analyse est applicable. Enfin, nous définissons le contexte de viabilité des connaissances produites $\mathcal{C}\kappa$, qui représente le domaine de validité des résultats produits.

Définition 3.9 (Contexte)

Nous définissons le contexte $\mathcal{C}$ comme :

$$\mathcal{C} = \mathcal{C}p \cup \mathcal{C}u \cup \mathcal{C}\kappa \tag{8.14}$$

Avec $\mathcal{C}p$, le contexte pédagogique, $\mathcal{C}u$ le contexte d'utilisation de l'analyse et $\mathcal{C}\kappa$ le contexte de viabilité des connaissances produites.

Cette structuration du contexte permet d'intégrer directement le contexte dans la narration des processus. Par exemple, par l'intermédiaire des éléments narratifs, il est possible d'expliciter l'influence des traces sur la définition de certains contextes de l'analyse. De plus, cela nous permet d'indiquer les points critiques du processus d'analyse narré, et de conditionner des comportements d'opérations. Reprenons l'exemple présenté dans la Section 8.1 à propos des comportements des opérateurs narrés sur le quotient et la moyenne. Contextualiser le fait qu'il s'agisse de notes issues d'un EIAH permet de conditionner la production de l'opérateur de division dans ce contexte précis, et ainsi remplacer dans le graphe de variables de sortie la variable quotient par une nouvelle variable *moyenne des notes d'un étudiant* représentative du contexte.

Afin de formaliser toutes ces parties, et notamment comment un graphe de variables d'entrée $\mathcal{G}e$ est manipulé par un opérateur narré $\mathcal{N}o$ pour produire le graphe de variables de sortie $\mathcal{G}o$ associé, ainsi que l'impact du contexte dans la définition des nouvelles variables, nous présentons notre Algorithme 2.

Algorithm 2: Transition d'état des variables et instauration des spécificités contextuelles

Input: e : l'étape courante
$\mathcal{P}e, \mathcal{P}p, \mathcal{P}o$: patrons de l'opérateur narré $\mathcal{N}o$ de l'étape e
$\mathcal{G}e$: le graphe de variables d'entrée
$\mathcal{C}$: le contexte de l'analyse
Output: $\mathcal{G}o$: le graphe de variables de sortie

Function calculGrapheSortie$(e, \mathcal{P}e, \mathcal{P}p, \mathcal{P}o, \mathcal{G}e, \mathcal{C})$
  $\mathfrak{V} \leftarrow \emptyset$
  $\mathfrak{V}' \leftarrow \emptyset$
  /* Procède à la mise en correspondance entre les variables d'entrée et celles du patron d'entrée. *Idem* avec les paramètres. */
  foreach $v_{\mathcal{P}e} \in \mathcal{P}e$ **do**
    $v \leftarrow FaireCorrespondreVariable(e, \mathcal{G}e)$
    $\mathfrak{V} \leftarrow \mathfrak{V} \cup \{v, v_{\mathcal{P}e}, w\}, w \in \mathcal{W}$
  foreach $v_{\mathcal{P}p} \in \mathcal{P}p$ **do**
    $v \leftarrow Specialisation(v_{\mathcal{P}p}, \mathcal{G}e)$
    if $v \neq \emptyset$ **then**
      $\mathfrak{V}' \leftarrow \mathfrak{V}' \cup \{v, v_{\mathcal{P}p}, w\}, w \in \mathcal{W}$
  $\mathcal{G}o \leftarrow \mathcal{G}e$
  $R \leftarrow \emptyset$
  /* Création des nouvelles variables et des relations qu'elles partagent avec les variables d'entrée utilisées, provenant du graphe d'entrée $\mathcal{G}e$ */
  foreach $b \in \mathcal{P}o$ **do**
    /* Identification des relations en lien avec b d'après les patrons $\mathcal{P}e$ et $\mathcal{P}p$. A est un ensemble d'arc $a = \{v_1, v_2, w\}$. */
    $A \leftarrow identifierRelations(b)$
    /* Définition contextuelle de la sémantique de la variable */
    $\mathfrak{b} \leftarrow Contextualisation(\mathcal{N}o, b, \mathcal{C}, \mathfrak{V}, \mathfrak{V}')$
    if $\mathfrak{V} \neq \emptyset \vee \mathfrak{V} \neq \emptyset$ **then**
      foreach $u \in \mathfrak{V}$ **do**
        $r \leftarrow DefinirTypeRelation(\mathcal{N}o, u, \mathfrak{b}, A, \mathcal{C}), r \in \mathcal{W}$
        $a \leftarrow \{v, \mathfrak{b}, r\}$, t.q. $u = \{v, v_{\mathcal{P}e}, w\}$
        $R \leftarrow R \cup \{a\}$
      foreach $u' \in \mathfrak{V}'$ **do**
        $r \leftarrow DefinirTypeRelation(\mathcal{N}o, u', \mathfrak{b}, A, \mathcal{C}), r \in \mathcal{W}$
        $a \leftarrow \{v, \mathfrak{b}, r\}$, t.q. $u' = \{v, v_{\mathcal{P}p}, w\}$
        $R \leftarrow R \cup \{a\}$
  /* Modification du graphe de variables de sortie en fonction des résultats $r \in R$ précédents. */
  foreach $r \in R$ **do**
    $V \leftarrow V \cup \{v_2\}$, t.q. $r = \{v_1, v_2, w\}, \mathcal{G}o = (\mathcal{V}, E)$
    $E \leftarrow E \cup \{r\}$, t.q. $\mathcal{G}o = (\mathcal{V}, E)$
  return $\mathcal{G}o$

Pour résumer l'algorithme qui représente la fonction d'état transition λ d'une étape, il faut tout d'abord que toutes les variables du patron d'entrée $\mathcal{P}e \in \mathcal{N}o \in e$ soient mises en relation avec les variables de $\mathcal{G}e$ qui doivent être manipulées par l'opérateur. Il en va de même pour les paramètres $\mathcal{P}p$. Ensuite, en accord avec le patron de sortie $\mathcal{P}o \in \mathcal{N}o$, le nouveau graphe de variables est produit. Il est défini à partir du graphe d'entrée et il est potentiellement altéré par des suppressions ou des modifications des variables déjà présentes, voire par des ajouts de nouvelles variables, le tout d'après les informations contextuelles adjointes. Il en va de même pour les relations.

8.4 Illustration

Dans cette section, nous illustrons la mise en œuvre de l'approche théorique de ce chapitre pour capitaliser les processus d'analyse. Pour ce faire, considérons l'exemple illustratif de la Section 6.1.2, page 83. En guise de bref rappel, il s'agit d'une analyse présentée par Agnihotric & *al.* (AGNIHOTRI et al., 2016) lors d'un workshop de la conférence EC-TEL 2016 (KLOOS et al., 2018). Elle se destine à prédire la certification des étudiants au sein de MOOC. En outre, cette illustration s'établit dans le cadre des étapes du cycle d'élaboration et d'exploitation de l'analyse, et de ses acteurs (cf. section 6.1, page 79). L'objectif est de montrer comment notre approche permet de décrire l'analyse d'après ces différents points de vue.

Nous présentons tout d'abord comment notre approche s'applique à concrètement décrire des traces puis, comment les étapes de l'analyse le sont. Enfin, nous considérons le processus d'analyse narré obtenu et explorons les possibilités narratives qui interviennent lors de sa finalisation, une fois que les étapes sont définies, et avant que les connaissances ne soient identifiées.

8.4.1 Narration des traces

Tandis que les outils d'analyse conventionnels utilisent des fichiers de données respectant un format spécifique, notre approche adopte une abstraction de ces données en variables (*i.e.* concepts des données véhiculées) sous forme de graphes de variables. Toutefois, dans tous les cas, la trace reste à l'origine de l'analyse et est modifiée au fur et à mesure des opérations. Aussi, il est fondamental d'en proposer le graphe de variables associé, qui servira alors de base à l'élaboration du processus d'analyse narré dans son ensemble.

Pour définir un graphe de variables initial approprié, il est donc nécessaire d'appréhender les spécificités de la trace identifiables lors de l'étape **E2** de sélection des traces pertinentes. Cela consiste à comprendre les données et les variables de la trace, mais aussi à comprendre et à expliciter les relations qui existent entre ces variables : l'expert de l'environnement d'apprentissage **R2** devient alors un élément crucial à la définition du graphe.

En effet, en reprenant le processus d'analyse proposé par Agnihotric & *al.* (AGNIHOTRI et al., 2016), l'on remarque qu'il est possible en observant les traces de conjecturer le sens de certaines variables, comme *course_id* ou encore *user_id*, directement à partir de leur nom. Cependant, il n'en va pas de même pour d'autres variables, comme la variable *LoE* – qui correspond au niveau d'éducation – où la sémantique est difficilement identifiable. L'objectif, lors de la définition d'un graphe de variables, est de définir ces variables avec des termes issus de notre vocabulaire sémantique. Par exemple, utiliser le terme *CAPTEN :MOOC* pour la variable *course_id* et *Apprenant_de_MOOC* pour *user_id*.

De plus, les relations qui existent entre ces variables sont absentes des fichiers conventionnels (*e.g.* CSV, JSON), alors qu'elles jouent un rôle majeur pour l'analyste : elles doivent donc être communiquées par l'expert de l'environnement d'apprentissage pour former le graphe proprement dit. Avec notre approche narrative, l'acteur peut ainsi intégrer directement dans le graphe de variables initiales l'ensemble de ces informations qui traduisent sa connaissance de l'environnement pédagogique.

La Figure 8.4 illustre comment le graphe de variables se construit itérativement à partir de l'intervention de l'analyste et de l'expert. Puisque la narration d'une analyse a vocation à être réalisée *a posteriori* de sa mise en œuvre, voire en simultané, l'expert, aussi bien que l'analyste, peuvent identifier les variables

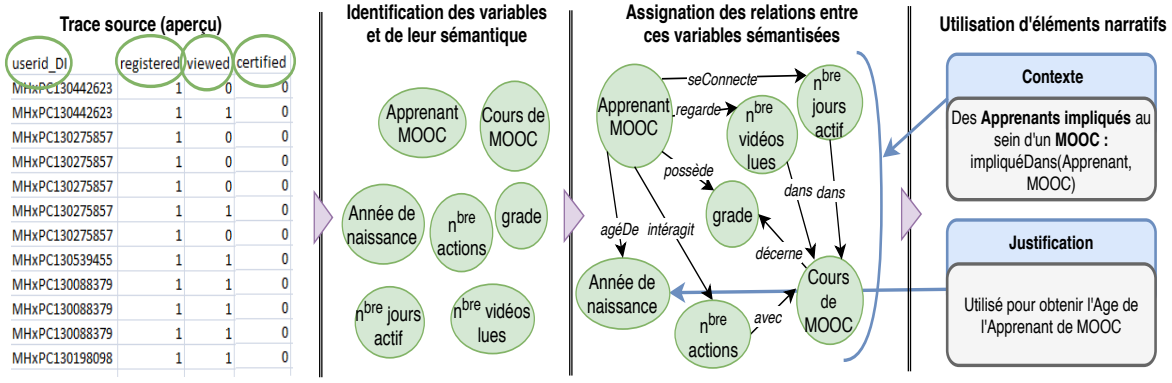


Figure 8.4.: Illustration de la procédure pour créer un graphe de variables avec notre approche narrative.

de la trace agrégée, ainsi que leur sémantique. Puis, les relations sont assignées à ces variables ; la présence de relations implicites tend à faire préférer l'expert pour cette tâche.

Enfin, les éléments narratifs et le contexte peuvent être intégrés dans la description du graphe de variables pour l'enrichir. L'expert de l'environnement peut comme cela expliciter le contexte d'où proviennent les variables représentées, mais aussi traduire l'interaction entre lui et l'analyste et les choix réalisés lors de la sélection de ces traces. Par exemple, la présence dans le graphe du nœud *Année_de_Naissance*[7] peut être justifiée – avec un élément narratif dédié à cette fonction – par le fait qu'il s'agisse de la seule source pour définir l'âge de l'*Apprenant_de_MOOC*, renforçant les possibilités d'adaptation.

8.4.2 Narration des étapes de l'analyse

Une fois les traces décrites, nous nous intéressons à la narration des différentes étapes qui vont constituer le processus d'analyse proprement dit. C'est principalement l'analyste qui réalise cette tâche puisqu'elle requiert une connaissance technique plus importante. En effet, il faut tout d'abord être en mesure d'identifier les raisons qui ont conduit à mettre en œuvre les manipulations opérées sur les variables afin d'identifier les étapes constitutives de l'analyse, au sens de notre approche. Ensuite, pour représenter une étape, il est nécessaire d'en identifier l'opérateur narré. Pour ce faire, l'analyste doit faire correspondre la sémantique de l'opérateur narré à celle de la manipulation ou à celle de l'ensemble des manipulations opérées dans l'étape. Enfin, il est nécessaire de décrire correctement le périmètre de cette étape : le graphe de variables d'entrée et les variables concernées par l'opérateur, les paramètres, les raisons de l'étape, le contexte, etc.

Un exemple de la création d'une étape a déjà été donné par la Figure 8.1 à propos de l'étape de corrélation. Il s'agit de la narration d'une des étapes du processus proposé par Agnihotric & *al.*. Nous y voyons le choix et l'utilisation de l'opérateur narré de corrélation, qui est mis en relation, par l'entremise de son patron d'entrée, avec des variables spécifiques d'un graphe de variable. Le graphe de variables de sortie représente les éléments résultant d'une telle application. Toutefois, bien qu'aucun élément narratif ni contextuel ne soit présent dans cette figure, il est tout à fait possible d'enrichir cette description ; et même recommandé !

En effet, prenons la première étape effectuée dans le processus décrit par Agnihotric & *al.*. Il s'agit simplement de l'obtention de l'âge de l'Apprenant de MOOC. Pour ce faire, les auteurs expliquent dans la documentation jointe que, puisque le MOOC date de 2013, il suffit d'effectuer la soustraction $2013 - x$, x étant la date de naissance de l'apprenant, pour définir l'âge. Dans cette première étape pourtant triviale où de nombreuses informations importantes sont présentes, très peu sont directement présentes dans le processus d'analyse – écrit en Python. Dès lors, pris indépendamment – rien ne précise par exemple à quoi correspond le nombre 2013 dans l'opération $2013 - x$ décrite dans le

7. Ce nœud représente la variable *YoB* du fichier CSV source.

processus : sémantiser cette information permettrait de renforcer la compréhension de l'étape et de faciliter son adaptation.

8.4.3 Narration de l'analyse

Pour finir, lorsque les différentes étapes constitutives du processus ont été narrées, il est encore nécessaire d'identifier les variables qui représentent les connaissances qu'il produit. L'analyste identifie ces connaissances dans les graphes de variables de sortie des étapes. Dans le processus de Agnihotric & *al.*, ces connaissances seront par exemple représentées par quatre classes d'apprenants – sémantisées – obtenues suite à l'application d'une opération de *clustering*.

Et finalement, il convient d'intégrer au processus d'analyse narré toutes les informations qui ne pouvaient être associées avant à des étapes particulières. Par exemple, le type de l'analyse représentée (*e.g.* prédictif, prescriptif), ou le besoin auquel elle répond. C'est donc ici que l'analyste explicite les informations globales sur l'analyse et son utilisation dans le cycle d'élaboration (*e.g.* les choix généraux d'implémentation, le contexte de validité de l'analyse, la manière d'utiliser les résultats et de les interpréter).

Narrer le processus d'analyse dans son ensemble offre aussi la possibilité aux décideurs d'expliciter les différentes informations en lien avec la formalisation du besoin, comme les considérations éthiques. Cela permet aussi aux bénéficiaires d'intervenir, et par exemple d'intégrer des informations à propos du contexte pédagogique spécifique à l'analyse, ou encore les éventuels retours d'utilisation. La Figure 8.2, page 113, offre un aperçu de cette démarche de narration du processus d'analyse une fois les différentes étapes représentées.

Ce qu'il faut retenir

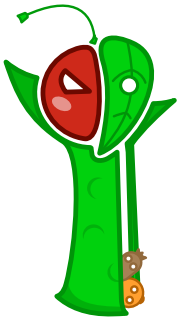
Dans ce chapitre, nous avons présenté l'approche **CAPTEN-ATOM** qui consiste à narrer les processus d'analyse de traces d'apprentissage. Nous avons aussi décrit sa partie théorique, à savoir **CAPTEN-ONION**, permettant :

- de s'affranchir des contraintes techniques (en renforçant l'approche présentée dans le Chapitre 7 d'après les résultats expérimentaux obtenus) ;
- d'enrichir la description des processus d'analyse de traces ;
- de structurer et d'intégrer les informations directement dans les processus (*e.g.* contexte pédagogique, choix d'élaboration) ;
- de sémantiser l'ensemble des processus décrits et leurs informations ;
- de couvrir les six propriétés identifiées comme nécessaires à la capitalisation ;
- de tenir compte des différentes étapes et acteurs du cycle d'élaboration et d'exploitation d'un processus d'analyse.

Cette approche s'appuie sur une ontologie qui définit sept notions majeures :

- celui d'**opérateur narré**, permettant de décrire de manière sémantisée le concept d'opération commun entre des opérateurs similaires mais implémentés dans des outils différents ;
- celui de **processus d'analyse narré**, permettant de décrire une analyse dans son ensemble, de manière indépendante des contraintes techniques et en intégrant et structurant l'information associée, nécessaire pour la capitaliser ;
- celui de **graphe de variables**, qui permet d'organiser les concepts présents dans les traces en fonction des relations qu'ils partagent entre eux ;
- celui d'**étape**, qui permet de représenter la mise en œuvre d'une opération dans le processus d'analyse, en plus de structurer les éléments y intervenant ;
- celui de **vocabulaire**, qui définit des classes et des propriétés sémantiques pour décrire tous les éléments utilisés dans le framework et lors de la définition d'un processus ;
- celui d'**élément narratif**, permettant d'intégrer l'information au processus d'analyse de manière structurée, notamment en attachant cette information aux éléments du processus et au processus lui-même ;
- celui de **contexte**, permettant de matérialiser le contexte et son impact dans le processus d'analyse.

9 Assister la réutilisation *via* une recherche intelligente par inférence sémantique



Sommaire

Section	**Introduction**	**129**
Section 9.1	**L'importance d'un nouveau mécanisme de recherche**	**130**
Section 9.2	**Description de l'approche**	**131**
Section 9.3	**Formalisation de CAPTEN-FRUIT pour l'assistance à la recherche**	**134**
Section	**Ce qu'il faut retenir**	**141**

Publications relatives à ce chapitre

(Lebis, 2018a) A. Lebis (2018a). "Assistance à la réutilisation de processus d'analyse de traces d'apprentissage via une approche narrative et sémantique". In : *Septièmes Rencontres Jeunes Chercheurs en EIAH (RJC EIAH 2018)*. Besançon, France

Introduction

Dans le Chapitre 8, nous avons présenté notre approche narrative visant à structurer sémantiquement un processus d'analyse de traces d'apprentissage ainsi que les informations associées, tout en les y intégrant. Les résultats expérimentaux que nous avons obtenus (cf. Partie IV, Chapitre 14) *via* l'utilisation par les acteurs de l'analyse de notre prototype (cf. Partie III, Chapitre 11) confirment que ce nouveau paradigme constitue une piste solide pour capitaliser les processus d'analyse.

Néanmoins, une conséquence inhérente à notre approche narrative est l'effort descriptif supplémentaire requis de la part des différents acteurs pour représenter un processus narré. En effet, chaque prise de décision, chaque information en lien avec l'analyse, peut être intégrée dans le processus *via* des éléments narratifs : cela requiert d'identifier l'élément narratif qui correspond le mieux ainsi que la relation partagée avec un élément de l'analyse, et de les décrire avec un maximum de termes issus du vocabulaire contrôlé. Ces phases de narration engendrent donc un coup non négligeable pour les différents acteurs impliqués dans la capitalisation des processus : il convient de présenter ici leurs bénéfices.

Ayant préalablement montré (cf. Chapitre 8) qu'une structuration sémantique de haut niveau de l'analyse est bénéfique pour la capitalisation, nous présentons dans ce chapitre une conséquence directe de la structuration de l'information des processus. Il s'agit d'une assistance intelligente pour la recherche des processus d'analyse de traces d'apprentissage. Cela consiste à exploiter les propriétés sémantiques existantes ou induites par notre ontologie – une fois peuplée – pour interpréter le besoin utilisateur et y répondre, par l'entremise d'inférences sémantiques, en proposant des processus d'analyse pertinents.

Aussi, dans la Section 9.1, nous présentons les motivations pour proposer une telle assistance. Dans la Section 9.2, nous décrivons les principes de notre recherche intelligente, avant de présenter dans la Section 9.3 les mécanismes d'ingénierie de connaissances qui la sous-tendent.

La mise en œuvre de cette proposition et les résultats associés sont présentés respectivement dans la Partie III, Chapitre 12 et dans la Partie IV, Chapitre 15.

Notons que notre approche narrative établit une base prolifique pour de nouvelles assistances. Bien que nous nous sommes particulièrement intéressés à fournir de nouvelles méthodologies de recherche - destinées non plus uniquement à l'analyse mais à tous les acteurs de l'analyse, d'autres pistes d'assistances sont évoquées en perspective, dans la Partie V. Elles permettent de présumer des différentes possibilités qu'offre l'intégration d'informations structurées et sémantisées au sein des processus d'analyse de traces d'apprentissage.

9.1 L'importance d'un nouveau mécanisme de recherche

Pour comprendre notre motivation à proposer une nouvelle approche de recherche des processus d'analyse, il est nécessaire d'examiner ce qu'implique la capitalisation des processus d'analyse de traces.

Comme nous l'avons vu, la capitalisation se constitue d'un ensemble de propriétés nécessaires pour permettre aux différents acteurs de pouvoir pleinement tirer profit des processus d'analyse existants. De prime abord, il ne s'agit alors que d'un ensemble de propriétés propres au processus, permettant de le qualifier comme capitalisable : les informations sont suffisantes pour permettre son adaptation et sa réutilisation à un nouveau besoin d'analyse. Nos travaux présentés dans le Chapitre 7 et le Chapitre 8 contribuent à fournir ces propriétés. Il s'agit du référentiel que nous avons adopté jusqu'à présent – celui des processus d'analyse.

Toutefois, la capitalisation s'observe aussi selon un autre référentiel : celui des acteurs de l'analyse ; puisque sans eux, point de besoin de réutilisation. Dans ce référentiel, la capitalisation s'exprime alors comme un ensemble de processus d'analyse qui coexistent et qui constituent un support potentiel pour répondre aux besoins d'analyse. Cela induit qu'un tel ensemble se doit d'être manipulable, permettant par exemple aux acteurs d'explorer, de consulter et d'interpréter l'existant. Par conséquent, la capitalisation promeut indubitablement des assistances à l'interaction des utilisateurs avec cet ensemble.

Cela s'observe dans notre état de l'art (cf. Section 3.1, page 25), et plus nettement dans le Tableau 5.4 (cf. page 65). Plus les outils tendent vers une approche communautaire et plus il est possible de recenser des dispositifs destinés à assurer la consultation ou l'exploration des analyses entreposées. Ainsi, des outils communautaires comme Galaxy (Goecks et al., 2010) ou ^{my}Experiment (C. A. Goble et al., 2010) permettent aux utilisateurs de rechercher les processus par des moyens avancés, comme des filtres de recherche. Toutefois, la possibilité de rechercher des processus – d'une quelconque manière – est une composante commune à tous ces outils.

Mais, quand bien même la recherche des processus serait une propriété universelle des outils d'analyse, il n'en reste pas moins qu'elle est à chaque fois superficielle, en cela que le système qui la conduit ne raisonne pas directement sur le processus d'analyse. Il n'utilise que quelques métadonnées (*e.g.*

le nom) comme nous l'avons vu dans notre état de l'art. En outre, la recherche n'est le plus souvent offerte qu'à l'analyste. Elle est confrontée à des limites importantes liées aux contraintes techniques générées par le paradigme actuel des analyses.

C'est pourquoi nous pensons qu'il est important de proposer une nouvelle manière d'assister les utilisateurs dans la recherche des processus d'analyse. En effet, il s'agit de la première tâche qui confronte les acteurs à l'ensemble des processus d'analyse qui sont à leur disposition. Il est donc crucial de ne pas oublier que, par conséquent, la recherche conditionne les analyses que les acteurs seront amenés à consulter et à utiliser.

Il est donc nécessaire de fournir une recherche tenant compte des différents acteurs de l'analyse, et aussi des particularités imputables à la fois à l'analyse et à notre domaine des Learning Analytics. Le cadre narratif que nous proposons nous offre un cadre propice pour cela, notamment grâce à la sémantique forte qu'elle implique. De plus, le fait qu'il s'agisse ici d'une composante commune aux outils communautaires nous fournit un cadre pertinent pour présenter les apports de notre approche [1].

9.2 Description de l'approche

Ayant rappelé que les recherches actuellement disponibles dans les outils d'analyse sont superficielles et non contextualisées, principalement à cause du paradigme actuel de mise en œuvre des processus d'analyse, nous présentons notre troisième proposition. Elle porte sur l'utilisation de la description des processus d'analyse narrés pour réaliser une recherche intelligente. Autrement dit, que le mécanisme de recherche soit capable d'interpréter les processus d'analyse narrés et de les mettre en relation avec un besoin d'analyse. La Figure 9.1 résume notre mécanisme de recherche et ses différentes étapes, décrites ci-après dans cette section.

Aussi, nous ne cherchons pas à définir un résultat ou un ensemble de résultats d'après quelques métadonnées ou des opérateurs utilisés, considérés indépendamment de leur contexte d'utilisation. Notre recherche porte sur l'intégralité du processus d'analyse, sur la manière dont les informations sont décrites, et sur les propriétés sémantiques qui existent entre elles. De plus, nous plaçons l'utilisateur au centre de cette recherche, en lui donnant la possibilité de consulter la pertinence évaluée par le système pour chaque processus d'analyse narré vis-à-vis de son besoin d'analyse, et les explications qui permettent de comprendre les résultats proposés. Nous cherchons ainsi à permettre l'assistance des différents acteurs de l'analyse dans l'exploration des processus d'analyse narrés et l'identification de potentiels candidats à la réutilisation.

Nous définissons donc cette recherche comme la possibilité de décrire de manière structurée un besoin d'analyse [2] et d'opérer automatiquement des requêtes sur les différents éléments narratifs des processus d'analyse narrés afin d'identifier les processus d'analyse, ou *a minima* les étapes, pertinents à réutiliser. Il s'agit de pouvoir raisonner avec chaque élément constitutif du processus (*e.g.* un opérateur, une étape, une hypothèse), d'exploiter le réseau sémantique offert par notre ontologie (cf. Chapitre 8), et de calculer une valeur de pertinence pour le besoin d'analyse décrit, tout en spécifiant les contributions de chaque élément y participant.

Lorsque l'on observe certains travaux des Learning Analytics engagés dans une approche communautaire (Projet HUBBLE, 2016 ; LearnSphere, 2018[d] ; Apereo Foundation, 2016), il est possible de constater certains efforts concernant la manière de décrire le besoin d'analyse associé à une analyse. Mais paradoxalement, cette richesse inhérente aux besoins d'analyse ne se manifeste pas lors de la recherche du processus d'analyse et n'est pas prise en compte par le système. Aussi, les résultats d'une telle recherche peuvent difficilement répondre aux attentes initiales des acteurs, et peuvent requérir une étape d'interprétation importante : cela diminue les acteurs aptes à utiliser une telle fonctionnalité de recherche.

1. La recherche des processus d'analyse joue également un rôle central dans des assistances plus complexes, comme l'attestent nos perspectives (cf. Partie V).

2. Nous nous émancipons de cette manière des mécanismes où le besoin est exprimé par l'intermédiaire de cases à cocher ou d'un simple texte utilisé ensuite dans une requête.

Il nous a donc semblé important de pouvoir correctement exprimer les besoins d'analyse lors des recherches. Pour ce faire, nous définissons un besoin d'analyse comme un ensemble fini de **dimensions**. Chacune représente une propriété différente, comme le contexte du besoin, sa description, les variables initiales à analyser, ou encore d'éventuelles hypothèses ou des outils d'analyse spécifiques à utiliser. En outre, ces dimensions impliquent des règles de recherche comme nous le verrons après : à cet égard, elles ne sont pas définies par l'utilisateur lors de la recherche, mais bien en amont – potentiellement par le concepteur de la solution. Cet acteur est toutefois libre de choisir les dimensions qu'il souhaite utiliser parmi les dimensions existantes lorsqu'il définit ledit besoin. La partie haute de la Figure 9.1 schématise la décomposition du besoin d'analyse en n dimensions.

Chacune des dimensions constituant le besoin d'analyse est décrite grâce à des **tokens** : ce sont des termes (*i.e.* classes [3] et propriétés sémantiques) qui statuent d'un point spécifique de la dimension en question. Ces termes sont directement issus de notre ontologie et du vocabulaire contrôlé de notre approche narrative. Ces tokens sont donc porteur de sens, et peuvent être mis en relation entre eux, pour spécifier des relations précises entre les différents éléments qu'il est important de conserver lors de la recherche. Par exemple, un enseignant pourrait définir le contexte du besoin d'analyse qu'il recherche comme étant des apprenants impliqués au sein d'un jeu sérieux. Dans notre approche, cela revient à définir dans la dimension *contexte* un token *involved(Student, Serious_Game)*, composé des termes *involved* (une propriété), *Student* (une classe) et *Serious_Game* (une classe).

Par conséquent, nous pouvons aussi exploiter les propriétés sémantiques de notre ontologie avec les tokens (*e.g.* subsomption, transitivité) pour affiner la recherche de candidats et l'évaluation de leur pertinence, notamment en fonction des retours utilisateurs. Pour procéder à cette évaluation, nous projetons individuellement chaque dimension dans notre ontologie, avant de procéder à une agrégation des résultats obtenues – et de leur score, comme nous le verrons dans la section suivante. Nous appelons les règles qui régissent la projection d'une dimension un **patron de recherche**. Il se compose d'un ensemble d'heuristiques de haut niveau pour l'utilisateur, qui définissent quels éléments de l'ontologie doivent être requêtés pour couvrir au mieux la dimension en question. De plus, pour chaque élément, une valeur d'importance lui est attribué, pondérant *in fine* le score de pertinence calculé pour les résultats (cf. *Calcul de la pertinence* par dimension de la Figure 9.1).

Ces patrons de recherche nous permettent donc de définir les éléments qui contribuent le plus à une dimension du besoin d'analyse, ainsi que leurs contributions respectives. Concrètement, une projection isole chacun des tokens décrivant une dimension et le recherche dans les éléments indiqués par le patron de recherche associé (cf. partie centrale de la Figure 9.1). La contribution d'un token est assignée lorsqu'une correspondance avec un terme contenu dans les éléments à rechercher est identifiée, et est pondérée par l'importance de l'élément : si la recherche est infructueuse, alors la contribution est nulle pour le token. Nous exploitons de plus les propriétés sémantiques existant entre les termes et les éléments de l'ontologie, comme la transitivité, pour enrichir cette recherche.

Néanmoins, le fait de ne pas trouver un token ne signifie pas qu'un concept similaire n'existe pas dans les processus d'analyse à disposition : il se peut que l'utilisateur n'ait pas employé la bonne sémantique, ou alors que le processus n'ait pas été décrit entièrement ou correctement. En s'appuyant sur les travaux d'approximation sémantique menés dans Corese (Corby et al., 2006a ; Corby et al., 2004), nous définissons la notion de **termes similaires**. Un terme est dit similaire à un autre, selon un certain degré de similarité, lorsqu'ils partagent tous deux un même arbre taxinomique, ou lorsqu'ils sont indiqués comme tel au sein de l'ontologie (cf. la flèche verte *terme similaire* dans la partie centrale de la Figure 9.1).

Cela permet d'envisager une propagation de la recherche dans un espace de recherche plus important, et de proposer à l'utilisateur des solutions n'étant plus strictement égales à un besoin d'analyse, mais similaires – selon un certain degré. Lorsqu'un token est approximé lors d'une recherche, le degré de similarité influe alors sur sa contribution pour la réduire – puisqu'il ne s'agit alors pas du concept exact recherché par l'utilisateur. Notons de plus que, lorsqu'un token initial est substitué par un token similaire, ses propriétés sémantiques deviennent elles aussi exploitables dans la recherche.

3. Autrement dit, des concepts de l'ontologie.

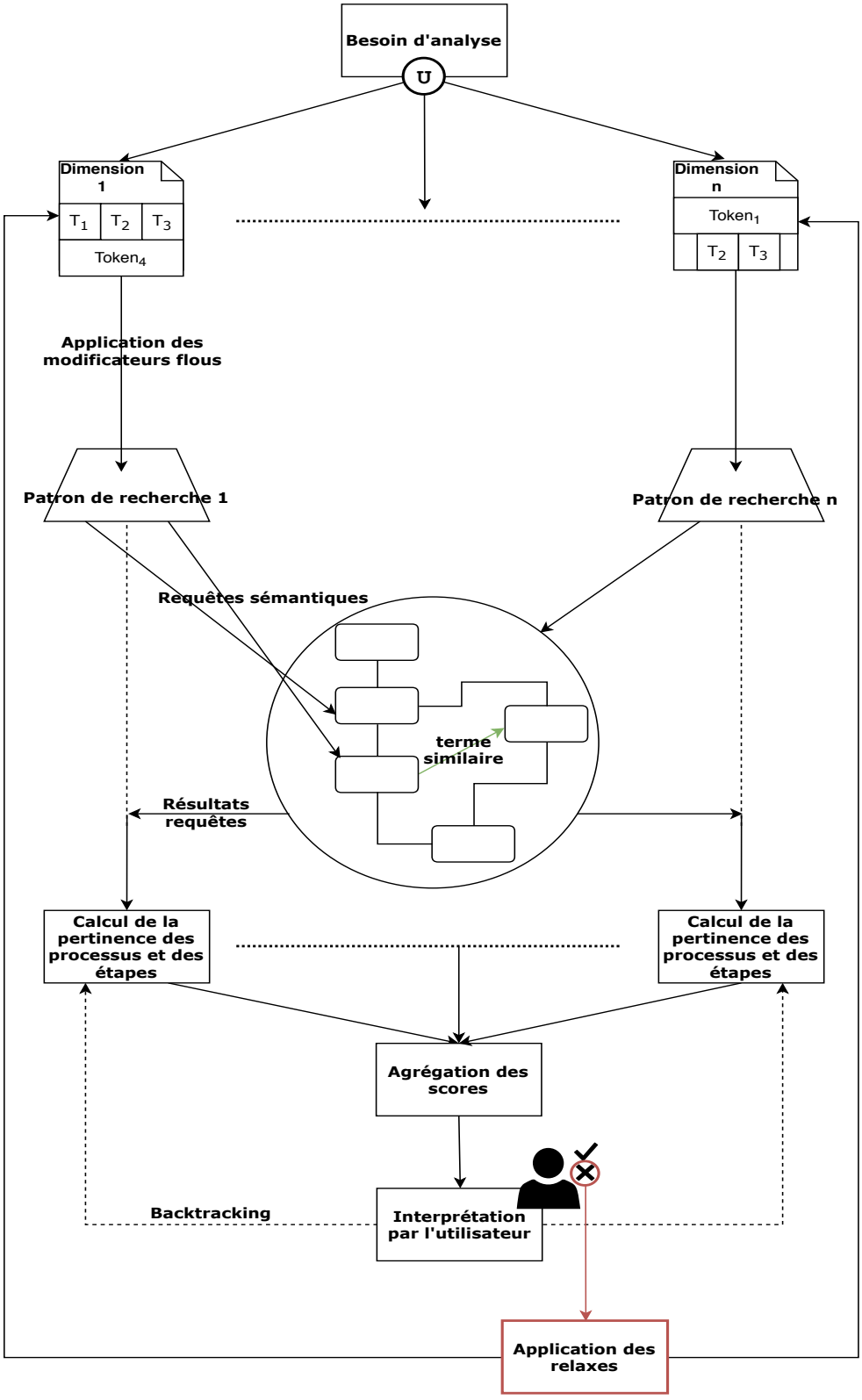


Figure 9.1.: Vue schématique de notre recherche intelligente fondée sur les informations et les éléments constitutifs des processus d'analyse narrés.

En outre, toujours soucieux de l'imprécision latente résidant dans la description d'entités (ici, du besoin d'analyse et des processus d'analyse), nous nous sommes inspirés des travaux en lien avec la logique

floue (ZADEH, 1971 ; ZADEH, 1965). Nous introduisons dans notre recherche un mécanisme d'approximation à l'échelle de l'utilisateur lorsque le besoin d'analyse est multi-tokens ou multidimensionnels. Aussi, lorsqu'une dimension possède plusieurs tokens, l'utilisateur peut exprimer l'importance qu'il leur donne suivant des **modificateurs flous**[4], comme *extrêmement similaire* ou *très peu équivalent*. Il en va de même pour chaque dimension du besoin d'analyse.

Toutes ces dispositions sont prises pour placer l'utilisateur au centre du mécanisme de recherche. En effet, lorsque le système requête l'entièreté des processus d'analyse existants, et calcule pour chaque processus sa pertinence, voire identifie potentiellement des étapes possiblement réutilisables, il établit une liste de résultats-pertinences.

De surcroît, pour assister l'utilisateur dans cette identification des candidats, nous avons doté notre recherche d'un mécanisme de **backtracking** (cf. partie basse de la Figure 9.1). Ce mécanisme permet d'identifier et de présenter à l'utilisateur quels éléments ont contribué au score de pertinence, pourquoi et avec quelle portée. Remarquons d'ailleurs que les outils actuels sont dépourvus d'explications concernant les résultats issus des recherches qu'ils mènent.

Enfin, dans le cas où les éléments identifiés ne conviennent pas à l'utilisateur, nous avons aussi intégré la possibilité de **relaxer les contraintes** inhérentes à la recherche des tokens, notamment celles associées aux tokens de relation. En effet, lorsqu'une relation de type $r(c_1, c_2)$ est recherchée, le système requête les différents éléments en quête de ce schéma, même dans le cas où le système est autorisé à utiliser la notion de similarité. Toutefois, il se peut qu'un tel schéma exact n'existe pas toujours. C'est pourquoi nous permettons à l'utilisateur de déclarer des concessions dans la description de son besoin, *a posteriori* de la première recherche. Cela permet de relancer une recherche, mais moins contrainte : les résultats auront potentiellement des scores de pertinence moins élevés, mais l'espace de recherche sera agrandi. Cela est illustré par la flèche rouge et les flèches rétroactives en bas de la Figure 9.1.

9.3 Formalisation de CAPTEN-FRUIT pour l'assistance à la recherche

Dans cette section, nous décrivons CAPTEN-FRUIT (in**F**erence **RU**les and heur**IsT**ics) : l'ensemble formel des différents éléments et procédés qui, une fois utilisés conjointement, permettent de fournir cette recherche intelligente.

9.3.1 Formalisation des éléments de la recherche

Avant de détailler l'algorithme de recherche, il convient de définir concrètement les différents éléments sur lesquels il repose. Nous formalisons ici le besoin d'analyse et sa représentation en l'adossant à notre approche narrative, et les heuristiques pour guider la recherche dans notre ontologie. En outre, nous expliquons comment s'opère la similarité entre les termes, et comment se définissent les relaxes, en plus de leurs impacts dans la définition du besoin d'analyse.

Décrire et subdiviser le besoin d'analyse : les dimensions et les tokens

Ancrés dans notre démarche narrative, nous avons modélisé le besoin d'analyse comme un ensemble fini de sous-propriétés. Chacune de ces sous-propriétés, que nous nommons dimension, doit posséder une sémantique propre ne recouvrant pas celle des autres. Cela nous permet d'attribuer des comportements de recherche spécifiques en fonction de ces dimensions, dans l'objectif d'obtenir à terme une projection la plus optimale possible. Actuellement, nous estimons à sept le nombre de dimensions pour couvrir un besoin d'analyse : le contexte ; l'objectif – ou *desiderata* ; la description ; les variables initiales à analyser ; les hypothèses ; les outils d'analyse spécifiques à utiliser ; le public cible.

4. Dans les premières itérations de notre approche, ces modificateurs flous s'assimilent plus à des coefficients de pondération appliqués au score de chaque token ou dimension, plutôt qu'à l'application d'une fonction de *fuzzyfication*.

Définition 3.1 (Besoin d'analyse)

Un besoin d'analyse $\mathcal{B}$ se définit par un n-uplet de dimensions $\mathcal{D}_i$, tel que :

$$\mathcal{B} = \langle \mathcal{D}_1, \ldots, \mathcal{D}_n \rangle \text{ , avec } n \in \mathbb{N} \tag{9.1}$$

La description des dimensions constitutives d'un besoin d'analyse s'opère uniquement par le biais de tokens. Un token est exprimé à l'aide des termes et des individus de notre ontologie et est soit un terme simple – représentant alors une classe au sens ontologique – soit une composition de termes qui s'exprime alors comme une relation. Nous n'autorisons donc pas la présence de textes libres dans la définition d'une dimension : là encore nous tâchons d'adopter une démarche narrative et contrôlée dans la description de ces dimensions. Cela s'explique puisque nous requêtons des éléments sémantiquement définis (*e.g.* un opérateur narré, une étape, une hypothèse) ; la présence de textes libres – et donc non interprétables par le moteur d'inférence sémantique – ne sera pas discriminante [5].

Définition 3.2 (Dimension et token)

Une dimension $\mathcal{D}$ se définit par un n-uplet de tokens T_i, tel que :

$$\mathcal{D} = \langle T_1, \ldots, T_n \rangle \text{ , avec } n \in \mathbb{N} \tag{9.2}$$

Un token T se définit soit par un 1-uplet lorsqu'il fait référence à un élément du vocabulaire contrôlé $\mathcal{W}$ (cf. Chapitre 8, page 121), tel que :

$$T = \langle w \rangle \text{ , avec } w \in \mathcal{W} \tag{9.3}$$

Soit par un 3-uplets lorsqu'il matérialise une relation spécifique à rechercher entre deux termes, tel que :

$$T = \langle w_1, w_2, w_3 \rangle \text{ , avec } w_{1,2,3} \in \mathcal{W} \tag{9.4}$$

Ce dernier token s'interprète alors comme l'existence d'une relation w_1 entre w_2 et w_3 et s'écrit : $w_1(w_2, w_3)$.

En guise d'exemple, supposons un acteur de l'analyse qui désire poser le besoin d'analyse suivant au système : "*On cherche à découvrir des régularités dans les actions des apprenants dans le contexte d'un jeu sérieux*". Dans notre approche, un tel besoin pourra être exprimé avec deux dimensions : la dimension de *desiderata* $\mathcal{D}_d$ et celle de *contexte* $\mathcal{D}_c$. Respectivement, la dimension de *desiderata* serait définie comme $\mathcal{D}_d = \langle découvre(Analyse, Actions), réaliséPar(Actions, Apprenant), possède(Actions, Régularité) \rangle$ et celle du contexte comme $\mathcal{D}_c = \langle Jeu\_Sérieux \rangle$

Les patrons de recherche : des heuristiques haut niveau pour l'utilisateur

Comme nous l'avons vu, nous proposons de guider la recherche en fonction des dimensions qui vont être recherchées grâce aux patrons de recherche. Chaque dimension possède un patron de recherche qui spécifie les éléments à consulter dans l'ontologie. L'objectif est d'une part de diminuer l'espace de recherche pour limiter la complexité et la combinatoire qui résulte du fait de requêter l'ontologie, et d'autre part d'identifier explicitement les éléments qui sont susceptibles de contribuer directement à chaque dimension [6]. Ils sont aussi conçus pour être facilement modifiables en ajoutant ou retirant des éléments de l'ontologie à visiter lors de la recherche.

5. Dans cette approche, nous n'avons pas encore exploré les techniques de traitement automatique des langues (TAL) ni les techniques d'indexation comme la Latent Semantic Analysis (LSA).

6. Dans les perspectives, nous expliquons comment ces patrons de recherche peuvent servir de socle à une approche par raisonnement à partir de cas pour affiner les éléments qui contribuent aux dimensions.

De plus, dans un patron de recherche, pour chaque élément de l'ontologie devant être requêté, un score d'importance lui est associé. Ce score, qui a pour but de figurer le degré d'importance de chaque élément dans la recherche, nous permet de pondérer l'importance d'un token trouvé : il permet de modérer l'impact lorsque le token a été trouvé dans un élément qui a été jugé comme ne contribuant que peu à la dimension en question, ou l'augmenter dans le cas d'un élément important pour la dimension. Aussi, cela illustre l'importance de définir des scores pertinents et significatifs lorsque les patrons de recherche sont élaborés.

Définition 3.3 (Patron de recherche)

Un patron de recherche $\mathcal{P}$ se definit par un n-uplet de couples élément-score tel que :

$$\mathcal{P} = \langle (e_1, s_1), \ldots, (e_n, s_n) \rangle \text{ , avec } n \in \mathbb{N} \tag{9.5}$$

Avec e un élément du framework ontologique $\mathcal{F}$, et $s \in [0; 1]$ un score d'importance, normalisé.

Suivant cette définition, nous proposons un patron de recherche associé à la dimension de *desiderata*, que nous avons utilisée lors des évaluations de notre mécanisme de recherche. Pour notre part, nous avons défini ces scores – et ainsi jugé de l'importance des différents éléments de l'ontologie pour cette dimension – empiriquement et itérativement *via* l'utilisation de notre prototype (cf. Chapitre 12, page 163). L'encart ci-dessous l'illustre et permet de servir d'exemple. L'hypothèse posée ici est que des informations en lien avec cette dimension se situent principalement dans des éléments de type *Objectif* et *Contexte* : lorsque des tokens sont identifiés dans ces éléments, l'on veut alors les favoriser. Les informations annexes (*Addendum*), ou encore les patrons de sortie sont aussi considérés comme des éléments pertinents pour cette dimension de *desiderata*.

Exemple 3.1 (Patron de recherche des *desiderata*)

$$\begin{aligned} \mathcal{P}_{\mathcal{D}_d} = \langle &(Connaissance, 0.9), (Objectif, 0.8), (Analyse, 0.9), (Addendum, 0.1), \\ &(Nom, 0.2), (Graphedevariables, 0.4), (Patrondesortie, 0.3) \rangle \end{aligned} \tag{9.6}$$

Les notions de similarité, de modificateur flou et de relaxe

Un autre point important que nous proposons est de traiter l'imprécision pouvant survenir lors de la description d'un processus d'analyse ou du besoin recherché. Pour cela, nous nous reposons sur les notions de similarité, de modificateur flou et de relaxe, qui sont des outils permettant d'étendre itérativement [7] l'espace de recherche de manière contrôlée. Ces aspects sont importants, puisqu'ils donnent aux utilisateurs un moyen direct d'interagir avec le moteur de recherche et offrent à ce dernier des moyens de pilotage de haut niveau : ils n'ont ainsi pas besoin de modifier les requêtes réalisées par le système.

Concernant la notion de **similarité**, elle peut s'opérer dans notre approche à deux moments distincts. Le premier est en amont de la requête de chaque token dans l'ontologie. Chaque token peut être modifié en un autre token d'après des règles de transformations précises (*e.g.* relaxes) et en consultant l'ontologie. Le deuxième est lorsque le raisonneur sémantique requête l'ontologie pour chaque token : les propriétés sémantiques existantes sont alors exploitées, pouvant amener à certaines conclusions ne correspondant pas exactement au token recherché (*e.g.* le token subsume un individu dans l'ontologie, et une propriété de transitivité est suivie).

Afin de gérer cette similarité en amont de la requête pour chaque token, nous utilisons un vecteur de similarité $\mathcal{V}s$ associé à chaque token. Pour définir les termes de l'ontologie qui seront insérés dans ce vecteur – entendez donc similaires au token – nous exploitons la taxonomie de l'ontologie ainsi que

7. C'est principalement l'utilisateur qui décidera s'il convient ou non d'étendre l'espace de recherche suite aux premiers résultats.

des relations sémantiques de similarité *ad-hoc* (*i.e. isSimilar*) qui sont utilisables lorsque le vocabulaire contrôlé est enrichi. Les éléments jugés similaires par le système sont ordonnés de manière décroissante dans le vecteur, suivant un score de similarité $\mathfrak{v}_t$. Ainsi, lorsque l'utilisateur autorise une recherche approximative selon un certain degré (c'est en partie l'utilité des modificateurs flous), une requête est effectuée pour le token et chaque terme similaire qui respecte le degré de similarité spécifié.

Définition 3.4 (Terme similaire)

Un terme t est dit similaire à un token T si :

$$\mathfrak{v}_t > 0, \mathfrak{v}_t \in [0;1] \tag{9.7}$$

Avec $\mathcal{V}s_T$ le vecteur de similarité pour T et $\mathfrak{v}_t$ le score de similarité de t avec T. Conséquemment, si $\mathfrak{v}_t = 0$, t et T ne sont pas similaires.

Il s'ensuit que chaque élément du vecteur de similarité $\mathcal{V}s$ est un couple $(t, \mathfrak{v}_t)$ traduisant, pour l'élément t, son degré de similarité $\mathfrak{v}_t$ vis-à-vis du token T.

Définition 3.5 (Calcul de la similarité)

À l'instar de Corese (Corby et al., 2006a), nous définissons la similarité entre deux termes d'après la distance ontologique d_H qui les sépare dans l'arbre taxinomique. Pour cela, la distance ontologique d_H entre deux termes t_1 et t_2 est définie comme le minimum de la somme des distances l_H de la chaîne de subsomptions entre les termes comparés et un ancêtre commun t, tel que (Corby et al., 2006b) :

$$d_H(t_1, t_2) = min_{\{t \geq t_1, t \geq t_2\}}(l_H(< t_1, t >) + (l_H(< t_2, t >))) \tag{9.8}$$

Avec la distance l_H entre deux termes t_i et t_j dans la chaîne de subsomption définie par :

$$\forall (t_i, t_j) \in H^2, l_H(< t_i, t_j >) = \sum_{\{t \in < t_i, t_j >, t \neq t_i\}} 1/2^{d_H(t)} \tag{9.9}$$

Lorsqu'une relation de similarité r_s est explicitement déclarée dans l'ontologie entre deux termes, la distance ontologique prend alors la valeur $\mathfrak{v}_{r_s}$ portée par cette relation.

$$\forall (t_i, t_j) \in \mathcal{W} \exists r_s, r_s(t_i, t_j) \Rightarrow d_H(t_i, t_j) = \mathfrak{v}_{r_s} \tag{9.10}$$

Toutefois, il n'est possible d'appliquer un vecteur de similarité que pour les tokens 1-uplet. La raison évidente étant qu'une mise en correspondance directe est réalisée. Pour les tokens 3-uplets, où il faut pouvoir tenir compte des trois éléments de manière distincte – notamment pour permettre les futurs relaxes, nous utilisons une matrice $4 \times n$, à 4 lignes.

Chacune des lignes permet de gérer respectivement la similarité de la relation entière $w_1(w_2, w_3)$, de la relation partielle $w_1(w_2, *)$, de la relation partielle $w_1(*, w_3)$ et enfin de la relation "atomique" w_1. La taille n de la matrice est définie d'après la relation générant le nombre maximum de couples $(t, \mathfrak{v}_t)$. Les autres lignes sont complétées par des couples $(\emptyset, 0)$.

Ensuite, nous proposons de moduler l'importance de chaque token, ainsi que de chaque dimension du besoin d'analyse, grâce à l'utilisation de **modificateurs flous**. Cette proposition est mue par deux objectifs : le premier est de donner à l'utilisateur la possibilité d'affiner sa recherche à la fois lorsqu'il la décrit et *a posteriori* des résultats ; le deuxième objectif est de fournir des modificateurs qui soient sémantiquement compréhensibles par l'utilisateur plutôt qu'une pondération numérique difficilement interprétable, notamment pour des acteurs dont l'expertise s'éloigne de l'analyse de traces, comme les enseignants.

Concrètement, nous appliquons une fonction de *fuzzification* [8] à chaque score de pertinence des tokens et des dimensions, que nous altérons par l'application du modificateur flou associé. Cela nous permet de modifier les conditions d'appartenance des tokens à l'ensemble solution (ZADEH, 1971 ; KERRE et DE COCK, 1999) et d'évaluer un nouveau score de pertinence, après *defuzzification* [9]. Une alternative est d'utiliser des coefficients de pondération associés à chaque modificateur flou, plutôt qu'une fonction, pour en modifier le score de pertinence [10]. Cela a l'avantage de ne pas requérir d'outils spécifiques à la logique floue pour réaliser la recherche.

Définition 3.6 (Score de pertinence et impact des modificateurs flous)

Le score de pertinence de réutilisation $\mathfrak{p}$ pour un élément résultat quelconque, noté e, par rapport à un besoin d'analyse lors d'une recherche tenant compte de l'imprécision imposée par les modificateurs flous m_i s'exprime comme :

$$\begin{aligned} \mathfrak{p}_e &= f_{m_j}(\mathfrak{p}_{\mathcal{D}_1}) + \ldots + f_{m_k}(\mathfrak{p}_{\mathcal{D}_n}) \\ \mathfrak{p}_{\mathcal{D}_i} &= f_{m_l}(\mathfrak{p}_{T_1}) + \ldots + f_{m_h}(\mathfrak{p}_{T_{n'}}) \end{aligned} \tag{9.11}$$

Avec $n, n' \in \mathbb{N}$, $m_{h,j,k,l} \in M$ l'ensemble des modificateurs flous disponibles.
D_i une dimension du besoin d'analyse et $T_{i'} \in D_i$, un token appartenant à une dimension du besoin d'analyse.
$\mathfrak{p}$ représente le score de pertinence d'une dimension ou d'un token, suivant l'indice associé.

Enfin, nous proposons d'intégrer au dispositif de recherche un mécanisme dit de **relaxe** : l'utilisateur peut explicitement choisir de relâcher des contraintes lors de la recherche concernant des tokens particuliers. Concrètement, en fonction de la relaxe, le système va modifier sa requête par rapport au token initial. L'utilisateur peut par exemple autoriser la recherche *via* tokens similaires. Cela nous permet de donner à l'utilisateur des leviers d'actions supplémentaires pour étendre, de manière contrôlée, l'espace de recherche. Toutefois nous estimons que, par défaut, une recherche doit être la plus proche possible de la description fournie par l'utilisateur : aucune relaxe n'est alors appliquée par défaut.

Nous avons défini onze relaxes différentes, réparties en fonction des tokens 1-uplet et 3-uplets. Dans le cas des tokens 1-uplet, les relaxes sont directes : soit il s'agit d'une mise en correspondance parfaite, soit l'on autorise l'utilisation de tokens similaires. Dans le cas des 3-uplets, nous proposons dix relaxes $\mathfrak{Q}$ différentes [11] :

— $\mathfrak{Q}_1$, la correspondance parfaite, soit $w_1(w_2, w_3)$;
— $\mathfrak{Q}_2$, correspondance du préfixe, soit $w_1(w_2, *)$;
— $\mathfrak{Q}_3$, correspondance du suffixe, soit $w_1(*, w_3)$;
— $\mathfrak{Q}_4$, correspondance du préfixe en suffixe, soit $w_1(*, w_2)$;
— $\mathfrak{Q}_5$, correspondance du suffixe en préfixe, soit $w_1(w_3, *)$;
— $\mathfrak{Q}_6$, correspondance de la relation qu'importe les classes, soit $w_1(*, *)$;
— $\mathfrak{Q}_7$, correspondance à la classe du préfixe, soit w_2 ;
— $\mathfrak{Q}_8$, correspondance à la classe du suffixe, soit w_3 ;
— $\mathfrak{Q}_9$, correspondance aux classes, soit $w_2 \sqcup w_3$;
— $\mathfrak{Q}_{10}$, correspondance séparée, soit $class(w_1) \sqcup w_2 \sqcup w_3$ [12].

8. Quantification floue.
9. Clarification de la quantification floue.
10. Cela est assimilable à une fonction de seuillage f où, pour tout $x < x_{modificateur}$, $f(x) = 0$, sinon $f(x) > 0$, tel que $f(x) < scorePertinence$.
11. Le symbole * s'apparente à un joker et signifie *n'importe quel élément*.
12. Il est ici nécessaire de trouver une similarité entre la relation w_1 et une classe de l'ontologie.

9.3.2 Formalisation de la recherche basée sur l'inférence sémantique

Dans notre recherche d'éléments à réutiliser pour répondre à un besoin d'analyse, nous l'avons vu, nous ne considérons que les opérations narrées (*i.e.* opérateurs et processus d'analyse) et les étapes comme de potentiels candidats à la réutilisation. De ce fait, lorsqu'une requête pour un token trouve un élément de l'ontologie, nous identifions à quelle opération ou étape l'élément appartient, et nous le considérons alors comme une potentielle solution en lui associant le score de pertinence lié au fait d'avoir trouvé ce token dans notre ontologie.

La recherche s'opère donc en plusieurs étapes. Premièrement, le système requête l'ontologie [13], pour chaque token de chaque dimension, en accord avec les relaxes configurées. Suite à ces requêtes, nous définissons alors un vecteur résultat $\mathcal{V}r$ pour chaque token, qui contient tous les éléments e de l'ontologie où le token a pu être identifié, ainsi que le score $\mathfrak{p}$ associé d'après le patron de recherche : nous avons donc un ensemble de couples $(e, \mathfrak{p})$. Nous pondérons ce score en fonction de si un token similaire a été utilisé, ou si un modificateur flou est appliqué. Ensuite, nous agrégeons ces vecteurs résultats à l'échelle de la dimension afin d'obtenir la liste de tous les éléments contributifs à ladite dimension – et leur score.

Pour agréger ces vecteurs résultats à l'échelle de la dimension et ainsi obtenir la liste de tous les éléments contributifs, nous avons défini une opération transitive $\circledast$, qui se définit comme suit :

Définition 3.7 (Agrégation des scores de pertinences)

Soit $(e, \mathfrak{p})$ un couple résultat d'un vecteur résultat $\mathcal{V}r$, avec e un élément de l'ontologie et $\mathfrak{p}$ le score associé (par le patron de recherche). L'agrégation se formalise tel que :

$$\begin{aligned} &\text{Soit } (e, \mathfrak{p}_e) \in \mathcal{V}r, \\ &\forall e_i \in \mathcal{V}r_1, e_j \in \mathcal{V}r_2, \text{on a :} \\ &\bullet \text{ Si } e_i = e_j \text{ alors } (e_i, \mathfrak{p}_{e_i} + \mathfrak{p}_{e_j}) \in (\mathcal{V}r_1 \circledast \mathcal{V}r_2) \\ &\bullet \text{ Sinon si } e_i \neq e_j \text{ alors } (e_i, \mathfrak{p}_{e_i}) \in (\mathcal{V}r_1 \circledast \mathcal{V}r_2), (e_j, \mathfrak{p}_{e_j}) \in (\mathcal{V}r_1 \circledast \mathcal{V}r_2) \end{aligned} \tag{9.12}$$

Avec $\circledast$, une opération transitive qui, pour deux vecteurs résultats $\mathcal{V}r$, produit un nouveau vecteur $\mathcal{V}$ constitué de couples $(e, \mathfrak{p})$ définissant tous les éléments uniques des deux vecteurs résultats. Lorsqu'un élément existe dans ces deux vecteurs, le score de pertinence $\mathfrak{p}$ est alors la somme des scores de pertinence des deux couples.

Il suffit alors d'appliquer $\circledast$ à chaque vecteur résultat $\mathcal{V}r$ d'une dimension pour définir le vecteur de résultats agrégés pour la dimension en question. Ensuite, dans un deuxième temps, il est nécessaire d'identifier les parents de ces éléments lorsqu'ils ne sont pas soit une opération narrée, soit une étape, pour pointer vers des éléments réutilisables. Cela est possible car, comme illustré par le diagramme UML de notre ontologie (cf. Figure 8.3, page 116), nous avons une structure hiérarchique dans notre ontologie qui nous assure l'appartenance à une opération.

Aussi, nous définissons un vecteur $\mathcal{V}t$ des solutions, là encore qui est un ensemble de couples élément-score $(e, \mathfrak{p})$. Ce dernier est enrichi de tous les éléments terminaux (*i.e.* étapes, opérateurs narrés et processus d'analyse narrés) présents dans $\mathcal{V}r$. Pour les éléments non terminaux, nous les déréférençons pour identifier leurs parents jusqu'aux éléments terminaux. Lorsqu'une étape ou une opération doit être ajoutée à $\mathcal{V}t$, si elle y existe déjà, nous lui ajoutons le score $\mathfrak{p}$ associé, sinon directement le couple élément terminal – score .

Enfin, ces vecteurs $\mathcal{V}t$ de solutions servent de socle à l'établissement des scores de pertinence liés à la réutilisation de chaque opération et étape pour le besoin d'analyse décrit. Puisque disposant d'un vecteur $\mathcal{V}t$ de solutions pour chacune des dimensions du besoin d'analyse, nous pouvons construire le

13. Les requêtes SPARQL utilisées sont présentées dans l'annexe F

vecteur $\mathcal{R}$ des résultats finaux à l'aide de l'opération transitive $\circledast$, à l'instar des vecteurs de résultats agrégés.

Le seul prérequis est, avant de créer ce vecteur final, d'appliquer sur les scores de chaque couple du vecteur solution de la dimension, la pondération d'importance qui lui est associée par les modificateurs flous. Dès lors, cet algorithme nous octroie, représenté par ce vecteur $\mathcal{R}$, l'ensemble des opérations et des étapes qui sont jugées réutilisables par le système, suivant un certain degré de pertinence. Ce degré illustre la couverture du besoin d'analyse décrit par l'utilisateur, et est intrinsèquement lié aux nombres de tokens qui ont pu être évalués lors des différentes requêtes précitées.

Ces résultats doivent logiquement être présentés à l'utilisateur dans un ordre décroissant – et éventuellement catégorisé – puisque nous adoptons une démarche additive de la notion de pertinence. De cette manière, la recherche suggère à l'utilisateur des potentiels candidats à la réutilisation.

En conservant l'ensemble des vecteurs calculés, il est aussi possible d'indiquer les raisons pour lesquelles une opération ou une étape possède un certain score de pertinence – ce qui ce fait à l'aide de notre mécanisme de backtracking.

Et finalement, l'utilisateur a la possibilité de choisir comment les différents tokens qu'il a décrits doivent être pris en compte par le système grâce aux différentes relaxes. Ces relaxes constituent de nouvelles manières de requêter l'ontologie et donc, potentiellement de nouveaux résultats. Aussi, nous pensons qu'il est intéressant de pouvoir mettre les résultats issus des différentes relaxes en relation les uns avec les autres.

Puisque le procédé algorithmique décrit ci-dessus est identique aux différentes relaxes, lorsque plus d'une relaxe est utilisée, nous plaçons les résultats de ces requêtes dans une matrice M de taille $(m \times n)$, avec m le nombre de relaxes utilisé et n le nombre maximal d'éléments résultats trouvés. Les autres lignes sont complétées par des couples $(\emptyset, 0)$. Conséquemment, il est évident que nous pouvons obtenir, quelle que soit la ligne de la matrice M, les vecteurs résultats $\mathcal{V}r$ correspondant et finalement les vecteurs $\mathcal{R}$ des résultats finaux. En ordonnant la matrice des résultats finaux $\mathcal{M}_{\mathcal{R}}$ en fonction des opérations et des étapes, il devient possible de les comparer entre eux, en fonction des différentes relaxes utilisées.

Ce qu'il faut retenir

Dans ce chapitre, nous avons présenté l'approche **CAPTEN-AERIS**, ainsi que sa partie théorique **CAPTEN-FRUIT**, qui repose sur notre approche narrative. Elle permet de :

— réaliser une recherche intelligente des processus d'analyse, des opérateurs et des étapes ;

— exploiter l'approche narrative et prendre en compte les informations véhiculées au sein des processus narrés ;

— décrire explicitement un besoin d'analyse à rechercher, de manière contrôlée et sémantisée ;

— expliquer les résultats obtenus ainsi que leur pertinence à être réutilisés ;

— placer l'utilisateur comme acteur central de la recherche.

Pour cela, elle s'appuie sur sept notions :

— celle de **dimension**, permettant de diviser un besoin d'analyse en propriété unique ;

— celle de **token**, qui sont des classes et des relations sémantiques issues de notre ontologie, utilisées librement, et qui permettent de décrire les dimensions ;

— celle de **patron de recherche**, qui définit des heuristiques de recherche pour chaque dimension et d'importance pour les tokens ;

— celle de **terme similaire**, qui permet de modifier un token en un autre suivant certaines règles pour étendre l'espace de recherche de manière contrôlée ;

— celle de **modificateur flou**, permettant d'affiner l'importance des tokens et des dimensions lors de la recherche ;

— celle de **relaxe**, permettant de relâcher des contraintes lors de la recherche pour étendre l'espace de recherche de manière contrôlée ;

— celle de **backtracking**, permettant d'expliquer les résultats à l'utilisateur et les contributions de chaque élément trouvé.

Partie III

Mises en œuvre

Introduction & Plan de la partie

Dans cette partie, nous présentons la mise en œuvre de nos modèles et de nos contributions théoriques, exposées dans la Partie II, au sein de trois prototypes. Cette partie nous permet d'illustrer, une par une, la faisabilité technique de nos propositions, d'apporter des pistes pour leur mise en œuvre, en plus d'introduire des exemples de fonctionnement supplémentaires pour compléter les explications et les exemples théoriques précédemment fournis. De plus, cette partie nous permet d'introduire le socle technique des expérimentations que nous avons réalisées, puisque ce sont ces prototypes que nous avons utilisés dans les expérimentations décrites en Partie IV.

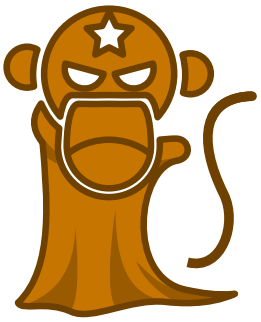
Aussi, nous présentons tout d'abord dans le Chapitre 10 **CAPTEN-APE**, le prototype mettant en œuvre notre proposition **CAPTEN-ALLELE** pour émanciper les processus d'analyse des dépendances techniques générées par les outils d'analyse.

Ensuite, dans le Chapitre 11, nous présentons notre prototype **CAPTEN-TORTOISE**. Il s'agit là de la réification de notre proposition théorique **CAPTEN-ONION** pour rendre les processus d'analyse capitalisables grâce à l'adoption d'un paradigme narratif pour les représenter.

Enfin, c'est dans le Chapitre 12 que nous présentons une mise en œuvre de **CAPTEN-FRUIT**, notre assistance à la recherche de processus d'analyse. Elle s'effectue au sein du au prototype **CAPTEN-SEED** et, après avoir présenté la manière dont il s'intègre à l'écosystème narratif, nous décrivons son fonctionnement, notamment en abordant différents cas d'utilisation. L'objectif ici est d'illustrer les mécanismes de recherche que nous avons présentés dans le chapitre 9 précédent.

Indépendance technique des processus d'analyse

10



Sommaire

Section 10.1 **Présentation du prototype CAPTEN-APE** **147**
Section 10.2 **Exemple d'utilisation de CAPTEN-APE** **149**

Publications relatives à ce chapitre

(Lebis et al., 2016) A. Lebis, M. Lefevre, V. Luengo et N. Guin (2016). "Towards a Capitalization of Processes Analyzing Learning Interaction Traces". In : *11th European Conference on Technology Enhanced Learning (EC-TEL 2016)*. T. 9891. Lecture Notes in Computer Science. Lyon, France : Springer, p. 397–403

(Lebis, 2016) A. Lebis (2016). "Vers une capitalisation des processus d'analyse de traces". In : *Rencontres Jeunes Chercheurs en EIAH (RJC-EIAH 2016)*. Montpellier, France

(Lebis et al., 2017b) A. Lebis, M. Lefevre, V. Luengo et N. Guin (2017b). "Capitaliser les processus d'analyse de traces d'e-learning". In : *Méthodologies et outils pour le recueil, l'analyse et la visualisation des traces d'interaction - ORPHEE-RDV*. Font-Romeu, France

10.1 Présentation du prototype CAPTEN-APE

Dans ce chapitre, nous présentons notre prototype **CAPTEN-APE** (**A**nalysis **P**rocesses **E**ditor) qui réifie tous les méta-modèles constituant notre proposition théorique **CAPTEN-ALLELE** (cf. Chapitre 7, page 93). **CAPTEN-APE** est une application web entièrement côté client, développée principalement en Javascript. Ce prototype est accessible en ligne (Lebis, 2018[c]), et une présentation en détail du prototype et de ses fonctionnalités est également disponible en ligne (Lebis, 2018[b]). Un aperçu de notre prototype est visible dans la Figure 10.1.

L'interface de notre prototype se divise en quatre zones principales d'utilisation, référencées (1), (2), (3) et (4) dans la Figure 10.1. La zone (1) regroupe les ressources qui sont à disposition de l'utilisateur lorsqu'il décrit un processus d'analyse indépendant. Ces ressources sont les variables et les listes initiales que l'utilisateur a pu déclarer à partir des traces qu'il possédait, les variables et les listes calculées automatiquement par le système et les opérateurs indépendants disponibles. Notons que les ressources calculées ne sont disponibles qu'*a posteriori* des étapes qui sont chargées de leur création.

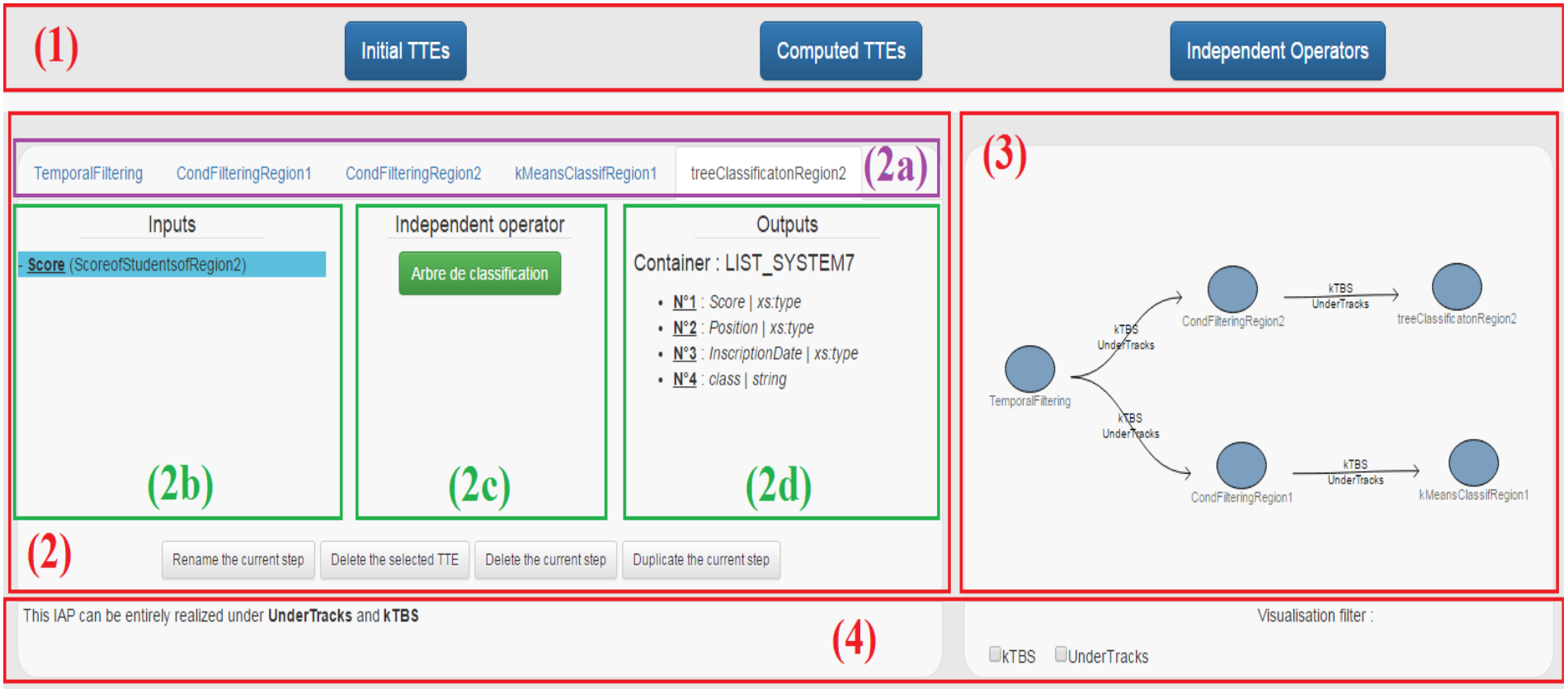


Figure 10.1.: Capture d'écran annotée de **CAPTEN-APE**, notre prototype web pour l'indépendance technique des processus d'analyse de traces.

Actuellement, treize opérateurs indépendants différents sont disponibles dans **CAPTEN-APE**. Nous avons identifié puis isolé les concepts d'opérations et les spécificités de ces opérateurs implémentés dans différents outils d'analyse [1], puis créé manuellement les opérateurs indépendants associés. Pour cela, nous nous sommes appuyés sur leurs fiches expérimentales et d'identification que nous avons dressées, dont un exemple est donné en Annexe A, Section A.1, page 234 et Section A.2, page 235.

La zone (2) permet à l'utilisateur de décrire un processus d'analyse indépendant. Dans cette zone, l'utilisateur crée une nouvelle étape pour représenter une opération à réaliser dans le processus : il y choisit un opérateur indépendant à utiliser, ainsi que les listes et les variables concernées. Aussi, la sous-zone (2a) est destinée à la navigation de l'utilisateur entre les différentes étapes – chaque étape constitue alors un onglet dans la zone ; la sous-zone (2b) contient la liste des variables qui seront utilisées dans cette étape spécifique ; la sous-zone (2c) indique l'opérateur indépendant utilisé ; et la sous-zone (2d) résume l'effet de l'application de l'opérateur indépendant sur les variables, en listant les nouvelles ressources qui seront créées.

Concernant la zone (3), il s'agit d'une représentation graphique du processus d'analyse indépendant inspiré des outils de workflows. Les nœuds du graphe représentent l'application d'un opérateur indépendant et les arcs illustrent l'ordre d'application de ces opérateurs et donc leurs interdépendances. Une dépendance intervient lorsqu'un opérateur indépendant est appliqué sur des variables qui sont le résultat de l'application d'un autre opérateur indépendant, sur d'autres variables. Enfin, les labels sur les arcs indiquent les éventuels outils d'analyse qui implémentent l'opérateur indépendant utilisé dans le nœud de départ.

Pour finir, la zone (4) est une zone informative qui a pour but d'assister l'utilisateur lorsqu'il envisage de répliquer ou de répéter son processus. Elle permet de lister les outils d'analyse capables d'implémenter entièrement le processus d'analyse indépendant décrit. Elle permet aussi d'appliquer des filtres de visualisation en fonction des outils d'analyse désirés, afin d'indiquer à l'utilisateur les différentes opérations qui peuvent être implémentées dans les outils choisis.

10.2 Exemple d'utilisation de CAPTEN-APE

Dans cette section, nous présentons comment l'indépendance technique des processus d'analyse est réalisée au sein de notre prototype. Pour cela, nous revenons sur l'illustration fournie dans la Section 7.4 de la Partie II, page 103. Il s'agissait d'y décrire un processus d'analyse existant et qui répond au besoin d'analyse suivant : "*Obtenir un indicateur relatant le pourcentage de visionnage d'une vidéo par une promotion d'étudiants sur un intervalle de temps, du 08/10 au 15/10*". Le processus d'analyse indépendant final obtenu est présenté dans la Figure 10.2.

Avant de décrire l'enchaînement des opérations qui doivent constituer le processus indépendant, il est nécessaire de décrire les variables initiales : celles qui sont primitives au processus d'analyse et qui y interviennent. Pour cela, il est impératif d'identifier dans les données du processus original les concepts qui ont été utilisés pour ensuite pouvoir les représenter en tant que variables dans notre prototype. Cette identification est pour le moment à la discrétion de l'utilisateur du prototype.

Une fois ces concepts identifiés, les variables doivent être créées dans notre prototype. Cela est réalisé grâce à une interface dédiée, présentée dans la Figure 10.3, qui permet à la fois de déclarer les listes initiales, ainsi que les variables qu'elles contiennent. Chaque variable est définie par un nom et un type. Dans cette capture d'écran, une variable *VisionnageEffectué* est créée, et définie comme étant un booléen. Elle est de plus associée à la liste *Trace Mooc Init*.

Une fois créées, les listes et les variables initiales sont disponibles à n'importe quel moment de la description de l'analyse. Pour utiliser les variables, il suffit de les ajouter à une zone de travail (*i.e.* les onglets dans la zone (2a) de la Figure 10.1) du processus d'analyse indépendant. Utiliser une variable revient aussi à impliquer la liste à laquelle elle appartient, et aide le système à calculer les variables potentielles en sortie.

1. Orange : DataMining, UnderTracks, R, RapidMiner et kTBS

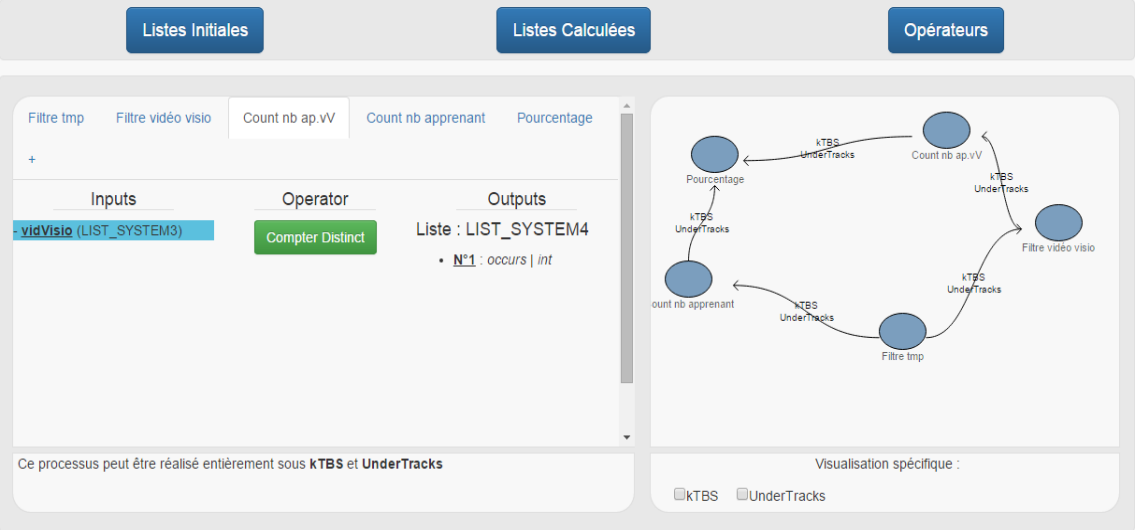


Figure 10.2.: Capture d'écran de **CAPTEN-APE** : un processus d'analyse indépendant pour le calcul du pourcentage de visionnage d'une vidéo.

Aussi, le processus d'analyse indépendant se décrit *via* l'application des opérations appliquées sur les variables. Par conséquent, chaque onglet de la zone (2a) fait intervenir au maximum un opérateur indépendant. Le nombre de variables sur lesquelles est appliqué l'opérateur dépend directement de ce dernier et de sa définition dans le prototype. La capture d'écran de la Figure 10.4 présente le menu de sélection des opérateurs indépendants, et permet de montrer comment un opérateur est présenté à l'utilisateur. L'on observe aussi diverses informations, comme son comportement (*OutputSheet*) ou encore les outils d'analyse qui implémentent cet opérateur indépendant. De plus, les opérateurs indépendants peuvent être configurés si des paramètres sont disponibles.

Lorsqu'à la fois les variables et l'opérateur indépendant ont été sélectionnés, le système calcule automatiquement les variables de sortie qui représentent l'application de l'opérateur sur les variables d'entrées. Elles permettent aussi d'apporter un retour utilisateur sur le comportement attendu du processus d'analyse. La Figure 10.5 montre ici l'application d'un opérateur pour compter le nombre d'occurrences distinctes d'une variable, ici du nombre de vidéos visionnées. Dans la colonne *Outputs*, l'on distingue la variable calculée *occurs* (du type entier) qui représente cette quantité.

Ces variables calculées sont ensuite utilisables dans les étapes suivantes de la description du processus, ce qui nous permet de définir l'évolution générale des variables. Comme expliqué précédemment, cette dépendance est matérialisée par les arcs du graphe (cf. Figure 10.1, zone (3)), qui se construit itérativement d'après les opérations ajoutées au processus. L'utilisation des variables calculées s'effectue de la même manière que les variables initiales, avec la contrainte qu'il n'est pas possible d'utiliser une variable calculée dans une étape antérieure à celle qui l'a créée. La raison est que nous filtrons dynamiquement les listes de variables calculées en fonction de l'étape courante de l'utilisateur : seules celles produites par des étapes qui n'ont pas de dépendance avec l'étape courante sont exploitables.

Ce choix de conception est notable puisqu'il empêche la description des processus d'analyse indépendants qui sont cycliques. En effet, nous pensons que ce type de processus d'analyse peut être défini à l'aide d'autres processus d'analyse, à l'instar de l'approche de découverte avec les modèles, proposée par Baker (R. S. J. D. Baker et Yacef, 2009). En sus, ce choix est également issu de la constatation empirique qu'aucun processus que nous avons rencontré n'arborait cette particularité : cela aurait alors rajouté une complexité supplémentaire non pertinente au prototype.

Enfin, notre prototype tient à jour en temps réel la zone d'assistance destinée à informer l'utilisateur des outils d'analyse capables d'implémenter le processus d'analyse décrit. L'utilisation des différents filtres de la zone (4) permet d'affiner l'assistance en vérifiant la couverture d'un outil d'analyse particulier à propos du processus : si elle est totale, partielle ou nulle – et les alternatives disponibles. De plus, le graphe est lui aussi mis à jour pour refléter la couverture de l'outil sélectionné.

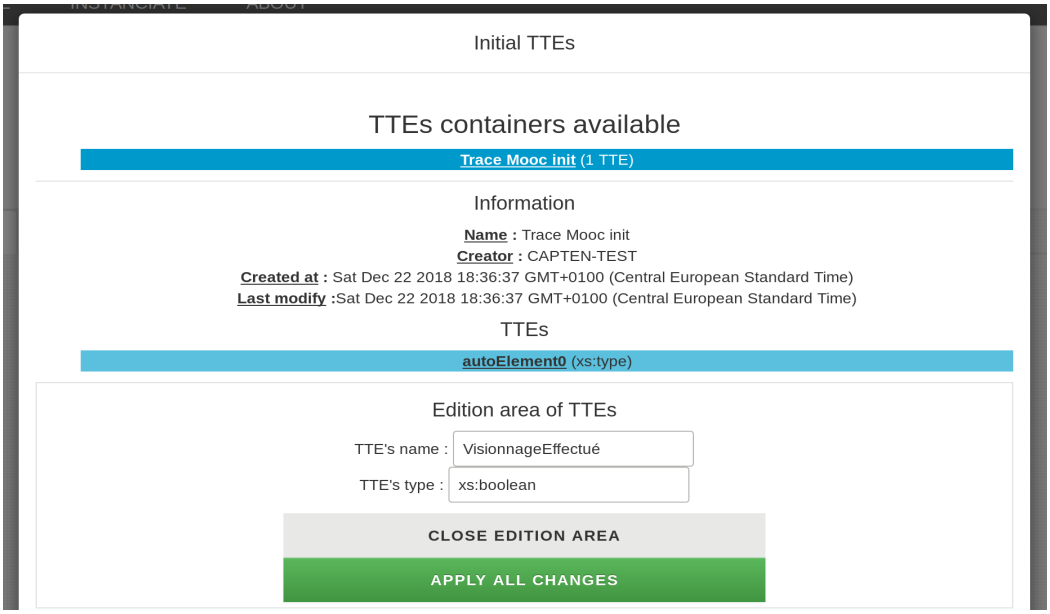


Figure 10.3.: Capture d'écran de **CAPTEN-APE** : page de création des listes et des variables initiales.

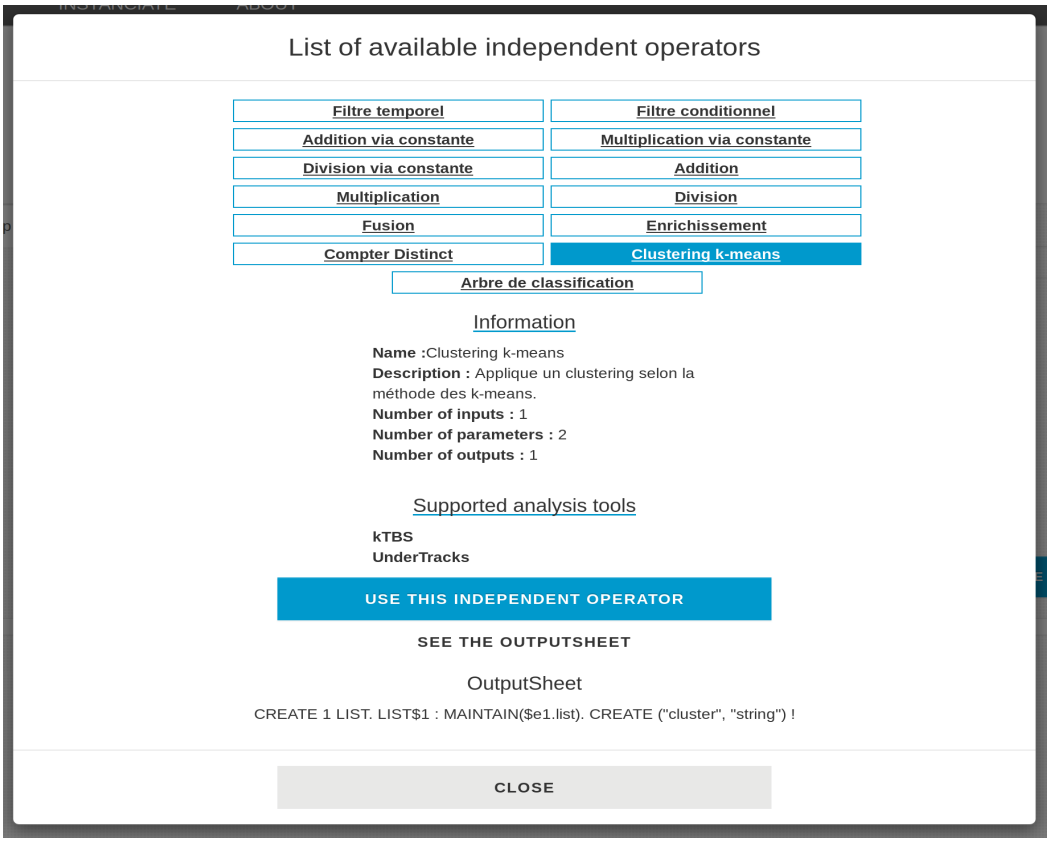


Figure 10.4.: Capture d'écran de **CAPTEN-APE** : choix d'un opérateur indépendant à utiliser dans une étape.

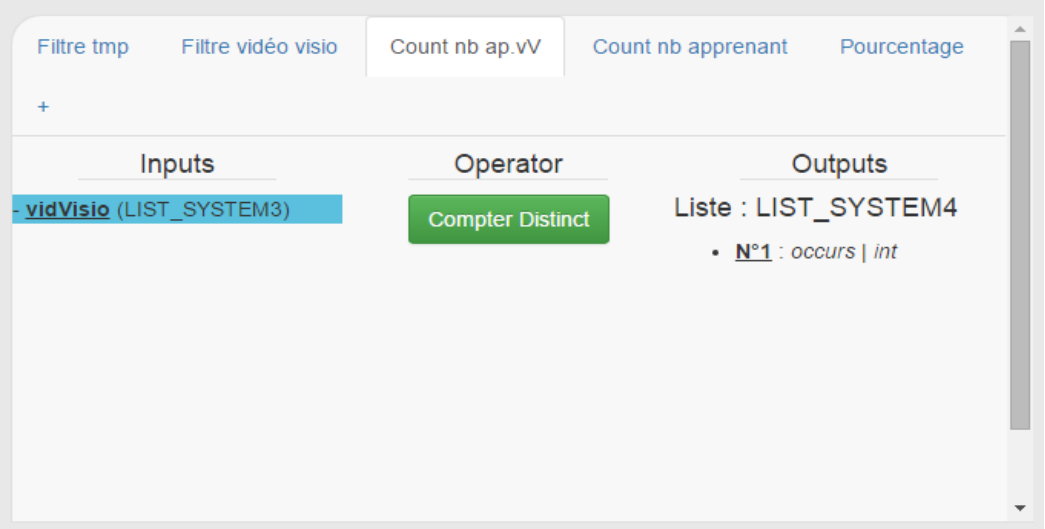


Figure 10.5.: Capture d'écran montrant comment les variables de sortie sont calculées dans notre prototype.

Narration des processus d'analyse pour les rendre capitalisables

11

Sommaire

Section 11.1	**Introduction**	**153**
Section	**Introduction**	**153**
Section 11.2	**Présention du prototype CAPTEN-TORTOISE**	**154**
11.2.1	Description des traces	154
11.2.2	Description des étapes de l'analyse	156
11.2.3	Narration de l'analyse	157
Section 11.3	**Peuplement du prototype avec l'existant**	**159**
Section 11.4	**Discussion**	**161**

Publications relatives à ce chapitre

(Lebis et al., 2017a) A. Lebis, M. Lefevre, V. Luengo et N. Guin (2017a). "Approche narrative des processus d'analyses de traces d'apprentissage : un framework ontologique pour la capitalisation". In : *Environnements Informatiques pour l'Apprentissage Humain*. EIAH 2017. Strasbourg, France

(Lebis et al., 2018a) A. Lebis, M. Lefevre, V. Luengo et N. Guin (2018a). "Capitalisation of Analysis Processes : Enabling Reproducibility, Openess and Adaptability thanks to Narration". In : *LAK '18 - 8th International Conference on Learning Analytics and Knowledge*. Sydney, Australia : ACM, p. 245–254

11.1 Introduction

Dans ce chapitre, nous présentons notre prototype **CAPTEN-TORTOISE** (narra**T**ive envir**O**nment for desc**R**p**T**i**O**n of analys**IS** proc**E**sses) développé pour capitaliser les processus d'analyse de traces d'apprentissage. Ce prototype met en œuvre la narration et implémente actuellement un sous-ensemble important de notre framework ontologique proposé dans le Chapitre 8. En effet, tous les éléments que nous avons présentés lors de ce chapitre, et qui sont illustrés dans la Figure 8.3 page 116, ont été implémentés au moins partiellement.

De plus, nous avons également entamé une phase de mise en correspondance de certains termes de l'ontologie chargée dans le prototype vers d'autres comme wf4ever (Page et al., 2012) ou encore xAPI (Advanced Distributed Learning, 2013) lorsque c'était possible. Par exemple, il est tout à

fait possible d'assimiler le concept de configuration d'un opérateur narré à celui de paramètre de wfdesc (wfdesc :Parameter)(BELHAJJAME et al., 2013c). En revanche, nous n'exploitons pas encore les propriétés intrinsèques de ces ontologies externes dans nos mécanismes de raisonnement (cf. chapitre suivant). Enfin, l'utilisateur peut lui aussi enrichir les termes disponibles avec ceux d'autres ontologies, comme nous le verrons dans la Section 11.2.1.

CAPTEN-TORTOISE est une application web développée en Javascript et utilisant principalement la bibliothèque Polymer et le paradigme de web components (GOOGLE, 2019). Bien que pensé pour être un hub communautaire où les différents opérateurs et processus d'analyse narrés pourront être partagés, il s'agit actuellement d'une application côté client. Son objectif est de permettre la description de ces opérateurs et processus d'analyse, ainsi que des étapes, de manière narrée et de favoriser leur adaptation et leur réutilisation. Il introduit également la notion de graphe de variables, de vocabulaire contrôlé et surtout d'éléments narratifs. Ce prototype est entièrement disponible en ligne (LEBIS, 2018[e]) et peut être utilisé localement [1].

Dans la section suivante, nous présentons en détail les différentes fonctionnalités du prototype, et comment un utilisateur peut décrire un processus d'analyse de manière narrée. Ensuite, dans la section suivante, nous expliquons comment nous avons décrit des processus d'analyse déjà existants dans notre outil, nous plaçant *de facto* en tant que fournisseur de processus d'analyse capitalisés. Enfin, dans la dernière section, nous faisons état de constats liés à cette étape de description des processus en processus d'analyse narrés.

11.2 Présention du prototype CAPTEN-TORTOISE

Dans cette section, nous montrons la mise en œuvre de l'approche théorique **CAPTEN-ONION** du Chapitre 8. Pour ce faire, nous reprenons l'exemple présenté dans la Section 8.4 illustrative, page 124, et nous conservons l'ordre dans lequel nous l'avons présenté : d'abord la narration des traces, puis des étapes et enfin de l'analyse. Son objectif est de permettre la prédiction de la certification des étudiants au sein d'un MOOC.

11.2.1 Description des traces

En accord avec notre approche narrative, les variables contenues dans les traces sont décrites sous forme de graphe de variables, où leurs relations sont représentées par des arcs. Pour définir le graphe de variables du processus d'analyse de la page 124, il convient d'étudier au préalable les données et les variables qui ont été utilisées, en plus des spécificités des traces qui ont pu être fournies directement par Harvard (MITx et HARVARDX, 2014). En effet, comme nous l'avons vu, ces traces contiennent des variables peu ou non sémantisées, comme *course_id* (pour l'identifiant du cours dispensé dans le MOOC) ou *LoE* (pour le niveau d'éducation de l'étudiant). De plus, les relations entre ces variables ne sont soit pas définies, soit expliquées dans des documents annexes.

Il est ensuite nécessaire d'opérer une mise en correspondance de chacune des variables impliquées dans l'analyse vers notre vocabulaire contrôlé pour identifier leur équivalent. Une fois que chacune des variables et des relations ont pu être identifiées de manière univoque, le graphe de variables représentatif est construit à partir de ces éléments équivalents. La partie supérieure de la Figure 11.1 montre, dans notre prototype, le graphe de variables initial décrit, et qui correspond aux variables nécessaires dans l'analyse proposée par Agnihotric & *al.*, page 124. Les nœuds et les arcs utilisés sont issus du vocabulaire contrôlé du prototype.

Une fois le graphe de variables décrit dans notre prototype, il est possible d'y attacher des éléments narratifs pour apporter et structurer de l'information supplémentaire. Cela s'effectue en choisissant le type d'élément narratif qui y sera associé, et en définissant son contenu à l'aide d'éléments du vocabulaire contrôlé et de texte libre. Pour ce graphe, les éléments narratifs que nous avons associés sont visibles dans la partie inférieure de la Figure 11.1. Nous y voyons par exemple un élément de

1. Les données utilisées pour peupler **CAPTEN-TORTOISE** peuvent être obtenues par demande directe auprès de l'auteur.

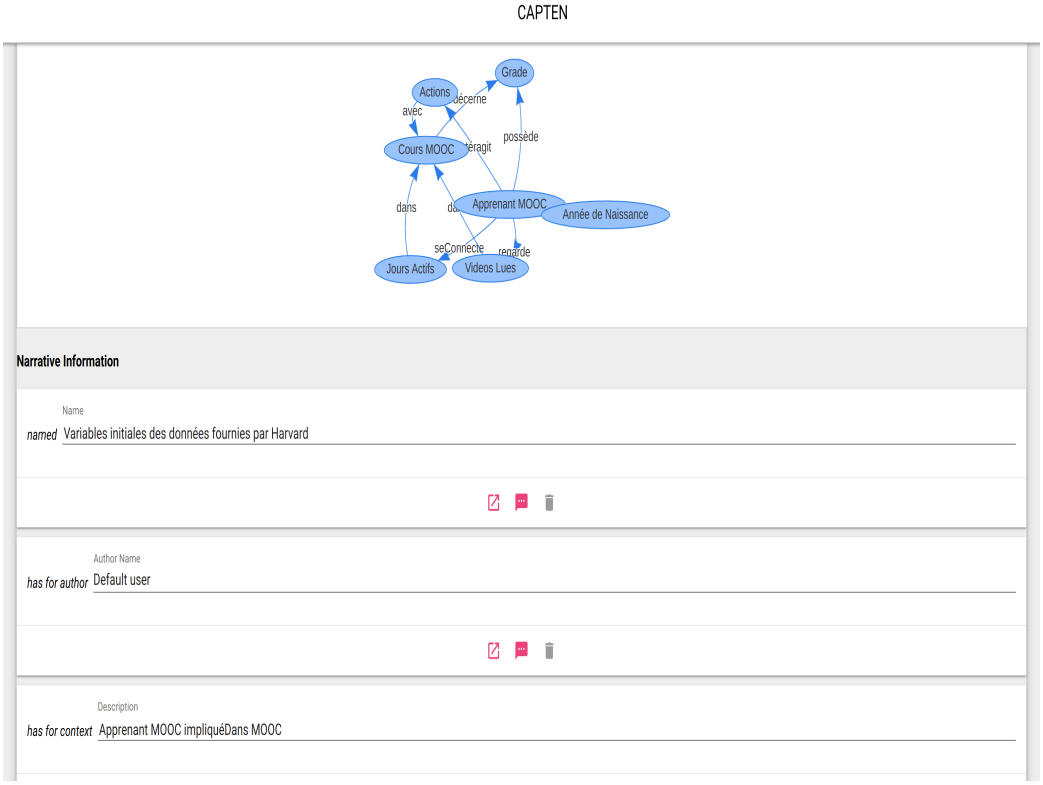


Figure 11.1.: Capture d'écran de **CAPTEN-TORTOISE** : page de description des graphes de variables.

type nom et auteur, comme l'on peut l'observer couramment. Nous y voyons également la présence d'un élément narratif de contextualisation, relié au graphe par la relation *hasContext* [2] et défini par quatre termes du vocabulaire : *Apprenant MOOC*, *impliquéDans* et *MOOC*.

Néanmoins, il se peut qu'aucun terme du vocabulaire ne convienne lors de la mise en correspondance des variables et des relations, ou lors de la narration. Toutefois, une solution de repli qui consisterait à utiliser du texte libre n'est pas forcément intéressante, puisque la sémantique est importante pour permettre au système de raisonner. C'est pourquoi il n'est, par exemple, pas possible de décrire les graphes de variables avec du texte libre.

Dans ces situations, nous proposons à l'utilisateur d'enrichir le vocabulaire disponible avec ses propres termes, qu'il pourra ensuite réutiliser lors de la description. Ici par exemple, nous avons ajouté au vocabulaire contrôlé le terme *Apprenant MOOC*. Cependant, si le nouveau terme n'est pas contextualisé avec le vocabulaire contrôlé, l'on conviendra facilement que le système ne pourra pas raisonner avec, cela revenant finalement à utiliser du texte libre. Aussi, en plus de permettre à l'utilisateur d'enrichir le vocabulaire, il est possible d'expliquer les propriétés que partagent le nouveau terme avec ceux déjà existants [3].

Dans la Figure 11.2, nous montrons comment cette contextualisation est mise en œuvre dans notre prototype. Ici, notre nouveau terme *Apprenant MOOC* est subsumée par la classe du vocabulaire *Apprenant*, déjà existante (assimilée au *Student* de xAPI). De cette manière, le système sait que la nouvelle classe partage les qualités d'un *Apprenant*, et cela pourra être exploitée lors des mécanismes de raisonnement.

2. Transcrite pour l'utilisateur comme *has for context*.
3. Actuellement, nous ne supportons que les relations de subsomptions. Les autres relations doivent être ajoutées manuellement.

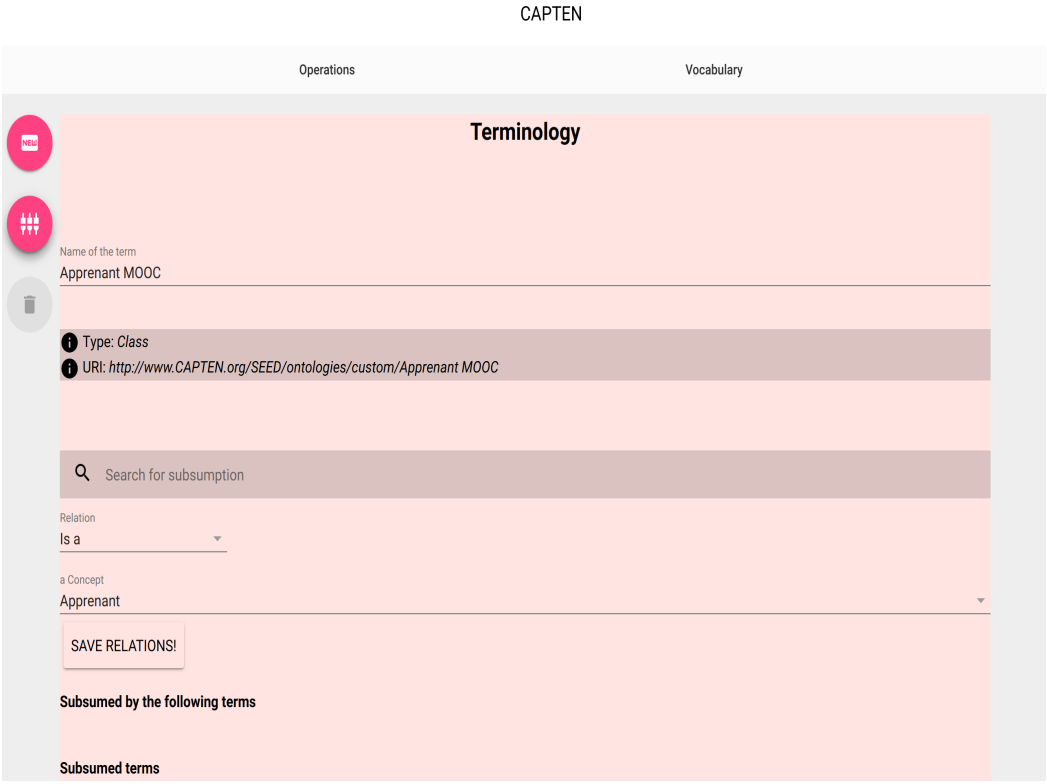


Figure 11.2.: Capture d'écran de **CAPTEN-TORTOISE** : page de mise en relation des termes du vocabulaire contrôlé.

11.2.2 Description des étapes de l'analyse

Il s'agit d'une mise en correspondance entre les étapes de l'analyse vers les futures étapes narrées. Puisque notre approche s'articule autour d'une approche atomique de l'intention d'analyse, cette mise en correspondance peut parfois être bijective, parfois surjective, parfois injective : cela dépend de l'outil d'analyse initial. Il est donc important d'identifier les intentions d'analyses, les traces, ainsi que les opérateurs utilisés et leurs configurations.

Intéressons-nous à comment décrire une étape dans notre prototype. Nous utilisons comme exemple l'étape de corrélation menée par Agnihotric & *al.* et que nous avons présentés dans la Figure 8.1, page 111. Il convient donc tout d'abord d'étudier cette étape. Nous voyons à la fois la transformation de la variable *YoB* (*i.e.* la date de naissance de l'apprenant) en âge et une corrélation appliquée sur cinq variables différentes, dont cette nouvelle variable âge. De plus, cette corrélation est réalisée par l'utilisation d'un opérateur python, configuré pour exploiter la métrique de Pearson. Il est également possible de relever dans les explications annexes la raison de la transformation de la date de naissance en âge [4], et l'objectif de cette corrélation.

Dans cette figure, l'on voit qu'il est possible de séparer en deux étapes distinctes – et correctement documentées – la transformation de la variable *YoB* et le calcul de la corrélation, mais que ces étapes sont dépendantes. En effet, la corrélation doit être appliquée sur l'âge des apprenants. Aussi, nous décrivons ici cette deuxième étape.

La description de cette étape narrée de corrélation est visible dans la Figure 11.3. Il s'agit de la page d'édition d'une étape dans notre prototype [5]. Pour décrire cette étape de corrélation, il est tout d'abord nécessaire de spécifier l'état des traces qui seront présentes en entrée par l'intermédiaire d'un graphe de variables. Ici, dans la partie *Input* située dans la zone à gauche de la figure, nous utilisons le graphe

4. Il s'agit, d'après les auteurs, du seul moyen d'obtenir l'âge de l'apprenant.
5. Une étape est toujours rattachée à un processus d'analyse, et ne peut exister par elle-même dans notre prototype.

de variables de sortie de l'étape précédente de transformation. De cette manière, nous avons alors à disposition l'âge plutôt que la date de naissance de l'apprenant pour réaliser la corrélation.

Ensuite, il est impératif d'indiquer l'opération à appliquer sur ce graphe de variables. Pour l'identifier correctement, il est nécessaire d'opérer une correspondance entre l'opération utilisée dans l'analyse à narrer et les opérateurs et processus d'analyse narrés, présents dans le prototype, qui respectent la noèse de son application. Là encore, cette mise en correspondance s'effectue manuellement. Dans notre prototype, cela revient à chercher l'opérateur ou le processus d'analyse qui correspond le mieux, et à l'utiliser. Nous utilisons ici un opérateur narré de corrélation [6].

Cette utilisation est visible dans la partie centrale de la Figure 11.3 où l'on voit l'opérateur narré de corrélation. Il est possible d'y discerner trois sous-parties : au centre à gauche, le patron d'entrée de l'opérateur narré ; au centre les informations narratives associées (*i.e.* son objectif), et ses paramétrages (*i.e.* le nombre de variables, le type de corrélation) ; au centre à droite, le patron de sortie (ici, une nouvelle variable sémantisée *correlation*). Le patron d'entrée doit ensuite être mis en relation avec les variables du graphe de variables d'entrée pour configurer correctement l'étape. Dans la figure, ces liens entre les variables du graphe et ceux du patron sont visibles dans la partie basse [7].

Lorsque l'étape a été correctement et entièrement configurée, le prototype calcule automatiquement le graphe de variables de sortie de l'étape. Autrement dit, le graphe d'entrée est enrichi de la variable présente dans le patron de sortie de l'opérateur narré, et est mise en relation avec les variables utilisées. Il est ensuite possible d'intervenir sur le graphe de variables de sortie en modifiant par exemple la sémantique des termes, voire des relations qui sont produites [8]. Dans cette étape par exemple, nous avons modifié la sémantique des relations pour faire apparaître la corrélation entre les différentes variables *via* la propriété *correléAvec*. La Figure 11.3 illustre tout cela.

Enfin, tout comme les graphes de variables (qu'ils soient d'ailleurs calculés ou non), il est possible d'attacher à cette étape des éléments narratifs pour structurer l'information. Par exemple, nous avons ajouté l'élément narratif *Objectif* pour expliquer que cette étape permettait d'identifier les variables potentiellement contributives à la prédiction. Nous avons également indiqué qu'il s'agissait d'une étape de prétraitement, et que le coefficient de Pearson utilisé était issu de l'hypothèse des auteurs de l'analyse qu'il est le plus couramment utilisé donc le plus pertinent. Cette narration se met en œuvre de la même manière que dans la Figure 11.1 pour la narration des graphes de variables, ou de l'analyse avec la Figure 11.4 suivante.

11.2.3 Narration de l'analyse

Pour finir, lorsque les différentes étapes constitutives du processus ont été narrées, il est nécessaire d'identifier lesquelles sont les étapes finales, à savoir celles qui représentent la production d'un résultat de l'analyse et donc d'une connaissance. Dans notre prototype, cela revient à identifier, dans le graphe de variables de sortie d'une étape, des termes comme des connaissances : le système considère alors l'étape comme finale, ce qui a pour effet de mettre à jour le patron de sortie du processus d'analyse s'il est amené à être utilisé comme un opérateur dans un autre processus.

Toutes les informations de description et d'explication qui ne pouvaient pas être associées aux étapes et aux graphes de variables doivent être intégrées au niveau du processus d'analyse en lui-même. La Figure 11.4 présente comment cette narration est mise en œuvre au niveau du processus d'analyse. Dans cet exemple, nous avons donc précisé pour le processus d'analyse narré son objectif, sa catégorie d'analyse (*i.e.* prédictif), le contexte dans lequel il est censé s'appliquer et le type d'analyse dont il s'agit.

6. Cet opérateur narré a été défini en amont lors du peuplement du prototype. Voir la Section 11.3 suivante.
7. La capture d'écran tronque cette liste de relations, n'y laissant que la première concernant l'âge lié avec la variable *Numeric Entity* du patron.
8. Une perspective serait d'enrichir le système avec ces particularités pour les prochains cas similaires.

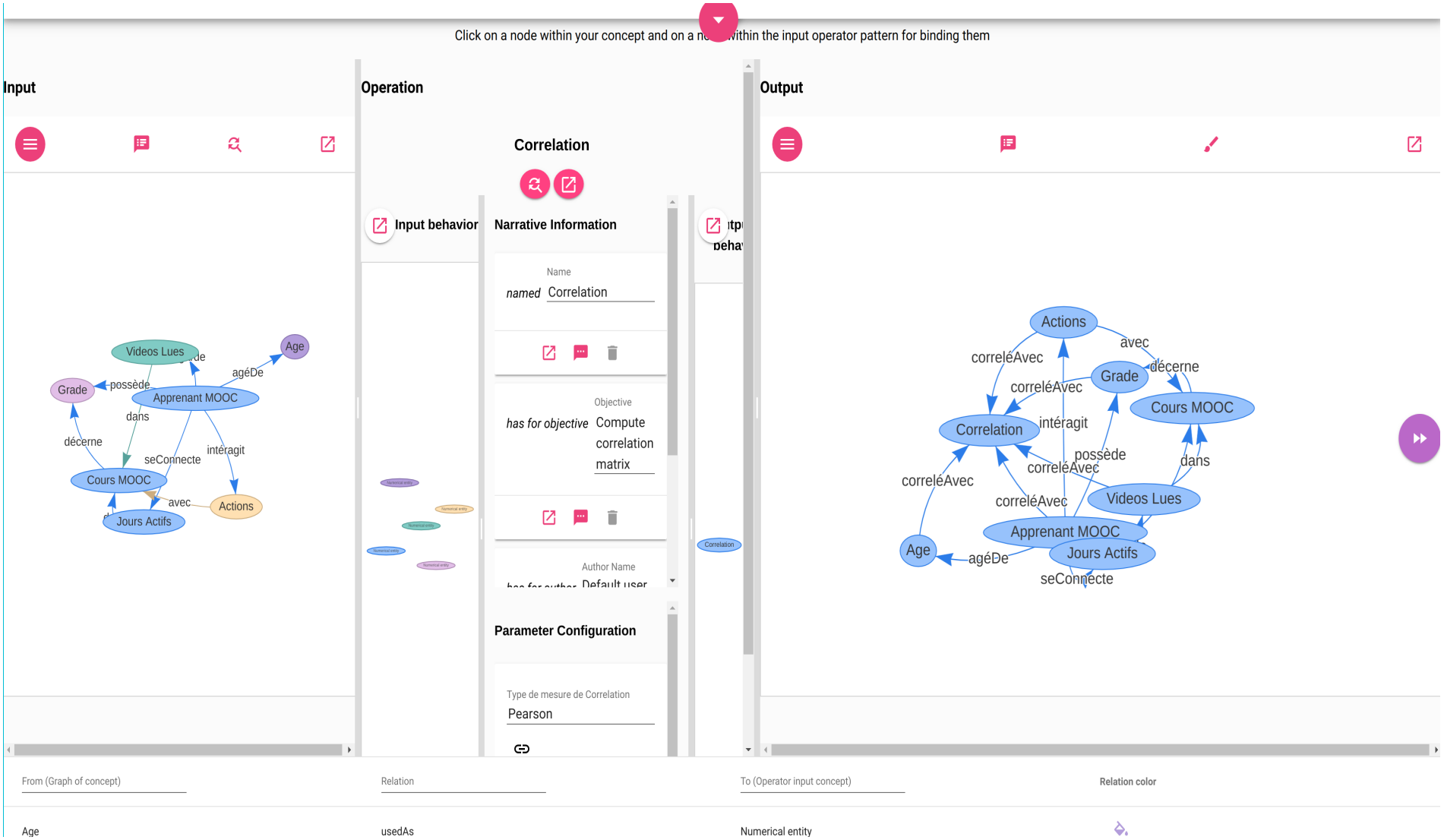


Figure 11.3.: Catpure d'écran de **CAPTEN-TORTOISE** : page d'édition d'une étape – entièrement configurée et où le prototype a calculé le nouvel état des variables.

Dans certains cas, les éléments narratifs peuvent être complexes et faire intervenir des dépendances avec d'autres éléments narratifs. Par exemple, une hypothèse est testée dans une étape de l'analyse, et est issue d'un article de recherche, dans un contexte spécifique. Notre prototype permet de réaliser ce type de description en adoptant une narration récursive. La seule limitation actuellement est que nous nous prémunissons des cycles dans la narration – notamment pour limiter la complexité lors du raisonnement par le système.

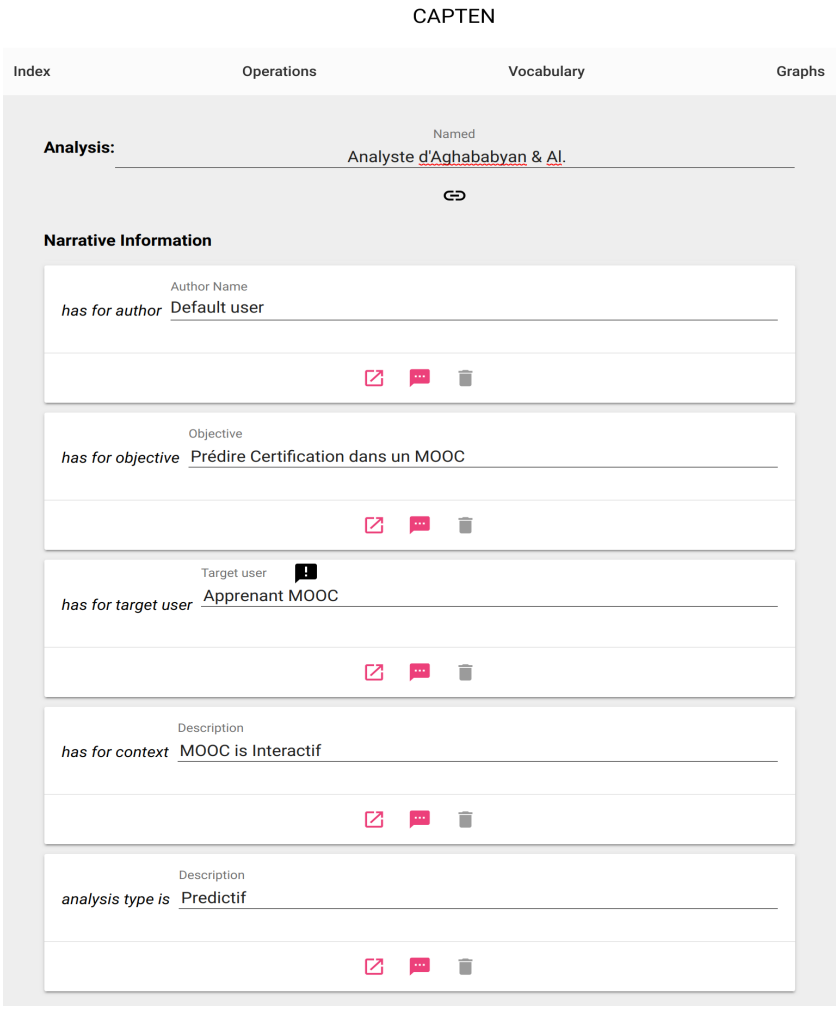


Figure 11.4.: Capture d'écran de la zone narrative d'un processus d'analyse narré.

En guise de conclusion, notons que certaines des fonctionnalités n'ont pas été évoquées en détail. Par exemple, il est possible, dans la vue du processus d'analyse, de consulter toutes les étapes et leur ordre d'application, à la manière d'un workflow. Il est aussi possible de créer un opérateur narré directement dans le prototype, et de lui associer un patron d'entrée et de sortie, et ainsi définir son comportement. Ou encore d'exporter et d'importer les différents éléments créés.

11.3 Peuplement du prototype avec l'existant

En prévision des expérimentations (cf. Chapitre 14, page 179), et pour évaluer notre prototype, nous avons procédé au peuplement de notre prototype. Avant d'importer des processus d'analyse existants dedans, nous l'avons tout d'abord enrichi avec des opérateurs courants et des termes courants dans notre domaine. Nous avons aussi étudié quelles terminologies issues des Learning Analytics et des EIAH il était intéressant d'importer en plus de celles déjà présentes (*via* l'utilisation d'ontologies externes comme xAPI). Pour ce faire, nous avons manuellement étudié des jeux de données et des

articles scientifiques afin d'extraire des terminologies récurrentes, comme *Prédictif* ou *Désengagé*. Nous les avons ensuite définies dans notre vocabulaire, avec l'interface présentée dans la Figure 11.2 et nous les avons enrichies manuellement lorsque nécessaire.

Après cela, nous avons consulté six outils d'analyse : Orange : Data Mining (Demšar et al., 2013), UnderTracks (Mandran et al., 2015), SPAD (Coheris, 2018), R (R Development Core Team, 2008), Knime (Berthold et al., 2009) et Weka (Witten et al., 2016). Lors de cette consultation, nous avons inventorié les différentes opérations que ces outils proposent, l'objectif étant d'identifier celles communes à une majorité d'entre eux. Nous avons alors défini ces opérations communes comme des opérateurs narrés dans notre prototype. Les trois patrons (*i.e.* d'entrée, de paramétrage et de sortie) ont été définis en étudiant le comportement de ces opérations dans leurs outils respectifs. Pour cela, nous avons observé les données que ces opérations attendaient en entrée dans chacun des outils, les données et les variables que ces opérations pouvaient produire et comment elles se configuraient – et l'impact de ces configurations. Puis, nous avons procédé à une identification manuelle de la sémantique de ces éléments, et les avons ensuite modélisés sous forme de graphes.

Nous avons ensuite choisi de décrire dans **CAPTEN-TORTOISE** neuf processus d'analyse issus du projet de recherche HUBBLE (Projet HUBBLE, 2016) principalement car nous avions accès à l'ensemble des documents décrivant ces analyses et que nous pouvions également discuter avec les utilisateurs les ayant élaborées [9]. De plus, ces analyses ont l'avantage d'avoir été réalisées *via* des outils d'analyses différents, certaines analyses en utilisant même plusieurs. Certains de ces outils étaient des tableurs (*e.g.* Excel), d'autres des outils orientés programmation (*e.g.* R), ou encore avec une approche workflow (*e.g.* Orange) ou bien des prototypes de recherche (*e.g.* UnderTracks, Usage Tracking Language).

Pour chaque analyse, nous avons premièrement identifié le besoin d'analyse auquel elle répondait pour le décrire dans le prototype à l'aide des éléments narratifs. Nous avons ensuite identifié les informations contextuelles (*e.g.* contexte pédagogique, bénéficiaires de l'analyse) en consultant les documentations et les données fournies. Nous avons aussi étudié précautionneusement les données initiales utilisées dans le processus d'analyse pour identifier les variables d'entrée du processus. Ensuite, nous avons extrait les relations existant entre ces variables, notamment grâce à la documentation fournie. Nous avons ainsi pu définir les graphes de variables d'entrée du processus, en utilisant l'interface dédiée (cf. Figure 11.1).

Ensuite, nous avons examiné chaque opération de chacune des analyses. Cela nous a permis d'identifier à la fois l'opération utilisée, mais également sa configuration, les données utilisées en entrée et les résultats produits. Cela nous a également permis d'identifier les intentions d'analyse mises en œuvre dans chacune de ces opérations. Aussi, nous avons extrait les différentes informations relatives à l'application de ces opérations dans les différents documents fournis. Nous avons ainsi pu procéder à la mise en correspondance de ces opérations vers les opérateurs narrés décrits précédemment dans notre prototype.

Cependant, il n'a parfois pas été possible de trouver dans **CAPTEN-TORTOISE** un opérateur narré pertinent pour décrire certaines opérations. En particulier lorsque l'opérateur narré qui aurait pu convenir pour décrire l'opération implémentée n'était pas utilisable du fait de spécificités techniques de l'outil d'analyse utilisé. Nous avons dans ces cas-là étudié les opérations précédant ou suivant l'opération à décrire, afin d'identifier un concept d'opération répondant à une intention d'analyse émanant de cette séquence d'opérations. Et c'est avec ce concept d'opération que nous avons ensuite procédé à la mise en correspondance avec un opérateur narré sémantiquement équivalent.

Nous avons ensuite défini chaque étape du processus d'analyse en utilisant un opérateur narré et mis en correspondance les éléments de son patron d'entrée avec les variables du processus. Le graphe de variables de sortie était automatiquement calculé par notre prototype (cf. Figure 11.3). Nous avons modifié ces graphes manuellement, lorsque nécessaire, pour affiner la sémantique générée automatiquement, et faire apparaître des relations plus représentatives de la sémantique des variables, en fonction des informations décrites dans la documentation de l'analyse. Par exemple, lorsqu'un opérateur narré de classification produit une répartition en classes, il est pertinent de préciser la nature

9. La liste de ces neufs processus d'analyse est disponible dans l'Annexe C.

des classes obtenues dans le contexte de cette analyse. Il est également parfois pertinent de préciser le sens des relations établies avec les variables du graphe de sortie calculé par **CAPTEN-TORTOISE**.

Après avoir ainsi décrit la structure en étapes du processus d'analyse, nous avons associé au processus et à ses étapes des éléments narratifs issus des informations que nous avions récoltées à partir de la documentation associée au processus et de l'étude des traces. Enfin, nous avons également identifié les connaissances amenées à être produites par chacune de ces analyses.

11.4 Discussion

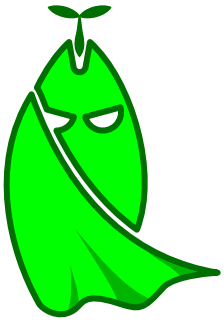
Tout d'abord, notons que nous avons réussi à décrire les neuf analyses au sein de notre prototype. Durant leur description, nous avons remarqué qu'il était possible de définir des sous-processus d'analyse, avec leur propre objectif et besoin d'analyse. Aussi, nous les avons décrits comme des processus d'analyse, que nous avons ensuite utilisés comme opérateurs dans la description des analyses principales.

Nous avons aussi rencontré des difficultés lorsqu'il s'agissait de décrire des opérations qui agissaient implicitement comme des opérateurs *pour chaque* ou *grouper par*. Nous suspectons que ce type d'opération est un agrégat complexe d'instructions qui véhiculent plusieurs objectifs en même temps. Néanmoins, nous n'avons pas réussi à trouver de dénominateur commun entre les différentes analyses et outils d'analyse. Pour pallier ce problème, nous avons donc décomposé ces opérations en sous processus d'analyse narrés, qui nous ont ensuite servi d'opérations.

De plus, il est important de noter que la qualité d'un processus d'analyse narré est implicitement liée aux efforts effectués pour le décrire. En effet, réifier un processus d'analyse existant dans notre prototype est une tâche qui prend beaucoup de temps et qui nécessite d'être méticuleux, puisque cela requiert à la fois de comprendre l'analyse et de s'émanciper des contraintes techniques induites par les outils utilisés. Il est nécessaire d'adopter une démarche que nous pouvons qualifier de déconstructive, puisque chaque élément de l'analyse en question doit être correctement associé à ceux de notre approche. Toutefois, nous nous attendons à ce que la charge d'efforts requis pour narrer les analyses soit moins importante lorsque les personnes qui ont participé à leur mise en œuvre s'en occupent.

Enfin, en décrivant ainsi ces neuf processus d'analyse dans notre prototype, nous avons pu montrer que la première partie de la capitalisation, décrite dans la Figure 6.4, page 90, est réalisable, à savoir formaliser des processus d'analyse à l'aide du formalisme de notre approche narrative. Ceci nous permet de valider que notre approche permet toujours de satisfaire les propriétés de réplicabilité, de répétabilité et d'ouverture nécessaires à la capitalisation.

12 Assistance *via* la recherche sémantique : CAPTEN-SEED



Sommaire

Section	**Introduction**	**163**
Section 12.1	**Présention du prototype CAPTEN-SEED**	**164**
Section 12.2	**Discussion**	**167**

Publications relatives à ce chapitre

(LEBIS, 2018a) A. LEBIS (2018a). “Assistance à la réutilisation de processus d’analyse de traces d’apprentissage via une approche narrative et sémantique”. In : *Septièmes Rencontres Jeunes Chercheurs en EIAH (RJC EIAH 2018)*. Besançon, France

Introduction

Dans ce chapitre, nous présentons notre prototype **CAPTEN-SEED** (**S**emantic s**E**arch **ED**itor). Ce prototype est une instanciation de **CAPTEN-FRUIT**, le mécanisme de recherche intelligente que nous avons présenté dans le Chapitre 9. Il implémente l’algorithme de recherche, et met en œuvre la majeure partie des notions présentées (*e.g.* la description du besoin d’analyse, les tokens, les patrons de recherche). Nous n’avons pas implémenté la propriété floue des modificateurs, nous reposant à la place sur des coefficients de pondération. De plus, dans ce prototype, le besoin d’analyse n’est constitué que d’après deux dimensions : l’objectif et le contexte.

CAPTEN-SEED est une application web côté client, développée principalement en Javascript. Il s’agit d’un greffon de **CAPTEN-TORTOISE** (cf. Chapitre 11), lui aussi accessible en ligne (LEBIS, 2018[e]) [1]. Les inférences et les requêtes sont réalisées en utilisant le raisonneur HyLAR (TERDJIMI et al., 2015), l’un des seuls raisonneurs Javascript à notre connaissance, ce qui nous permet de l’intégrer directement dans notre prototype. De plus, il est capable de raisonner à la fois côté client et côté serveur, et implémente un mécanisme de mise à jour des règles.

Dans la section suivante, nous présentons en détail les différentes fonctionnalités de notre prototype et comment un utilisateur peut moduler sa requête pour raffiner la description de son besoin d’analyse.

1. La page dédiée à la recherche des processus décrits dans le prototype est disponible à l’adresse suivante, lorsque le prototype est déployé en local : http ://adresse :port/#/needSearch

Pour cela, nous utilisons un exemple simple que nous appliquons sur l'ensemble des processus d'analyse que nous avons narrés au sein de notre prototype **CAPTEN-TORTOISE** (cf. Section 11.3, page 159).

12.1 Présention du prototype CAPTEN-SEED

Pour présenter notre prototype, et l'assistance à la recherche, nous adoptons un scénario d'usage autour d'un besoin d'analyse spécifique. Il s'agit de rechercher les processus d'analyse narrés qui proposent une classification des apprenants, le tout contextualisé dans le cadre d'un jeu sérieux. Dans notre approche et notre prototype, un tel besoin s'exprime avec la dimension de *desiderata* tel que $\mathcal{D}_o = \langle classify(Analysis, Apprenant)\rangle$, et la dimension contextuelle telle que $\mathcal{D}_1 = \langle Serious_Game\rangle$.

Toutefois, dans le cadre de cette présentation, nous avons expressément rajouté des contraintes à la dimension contextuelle, afin de n'obtenir aucune correspondance pour pouvoir présenter les relaxes de contraintes. Pour cela, la dimension contextuelle est définie comme $\mathcal{D}_c = \langle hasContext(Knwoledge, Serious\_Game)\rangle$ au sein du prototype. Notons que l'interface graphique mise à disposition de l'utilisateur ne permet pas encore d'exprimer des relations. Elles doivent être pour le moment ajoutées manuellement dans le fichier d'export correspondant.

Le besoin d'analyse ainsi défini, la Figure 12.1 représente les résultats obtenus après interrogation des processus narrés. Cette figure est constituée de deux parties : la première partie, labélisée par *Result display*, recense tous les éléments qui ont été identifiés comme de potentiels candidats, auxquels sont associés des scores de pertinence. Ici, un processus d'analyse narré intitulé *Classification des comportements utilisateurs* est estimé comme un résultat pertinent avec un score de 18 (nous reviendrons sur ce score). Dans la colonne *Token coverage*, l'on voit que ce résultat couvre les deux tokens qui constituent le besoin d'analyse.

L'utilisateur peut choisir de ne pas s'intéresser au détail de la recherche et de se contenter des résultats agrégés ou, au contraire, se renseigner sur les différentes dimensions et leurs contributions respectives, et ce par plusieurs moyens. Tout d'abord, il peut vérifier, pour chaque résultat, quels tokens ont été correctement répondus par dimension. La Figure 12.2 illustre pourquoi le système propose ce processus d'analyse narré spécifiquement. L'objectif ici est de fournir un ensemble d'informations susceptibles d'assister l'utilisateur dans l'identification de solutions potentielles.

Si nécessaire, et cela constitue la deuxième partie de la Figure 12.1, l'utilisateur peut consulter le détail de chacune des dimensions constitutives du besoin d'analyse. Cette deuxième partie liste toutes les dimensions, les tokens qui ont été recherchés par le système et les contraintes associées, et les résultats identifiés comme pertinents pour chacune de ces dimensions [2].

Dans la sous-partie *Dimension : Objective* de cette deuxième partie, nous voyons qu'un seul processus d'analyse a été identifié par le système comme potentiel résultat (il s'agit de l'analyse précitée). Dans la sous-partie *Dimension : Context*, nous pouvons voir qu'un résultat a été obtenu également, mais que celui-ci n'est pas issu d'une correspondance parfaite, comme c'est le cas pour la dimension de *desiderata*. En effet, nous avons relaxé les contraintes sur le token *hasContext(Knowledge,Serious_Game)* avec la relaxe $\mathfrak{Q}_8$ (cf. Section 9.3, page 134), pour indiquer au système que nous recherchions une analyse qui s'établit dans le contexte d'un jeu sérieux uniquement : nous ne la contextualisons plus en fonction des connaissances produites. Recourir à des relaxes permet ainsi de cibler un espace de recherche plus ou moins important en modifiant la manière dont le système interprète le besoin d'analyse de l'utilisateur, tout en permettant à ce dernier de le faire avec des mécanismes de haut niveau.

En outre, grâce à l'implémentation du mécanisme de backtrack, notre prototype permet d'expliquer à l'utilisateur pourquoi le résultat a été identifié comme tel. Cela lui permet de consulter les différents résultats et d'identifier lui-même s'ils conviennent à son besoin, notamment en consultant les éléments qui ont contribué à les identifier. L'utilisateur peut en effet voir quels tokens ont été trouvés lors de la

2. Chaque dimension possède un patron de recherche spécifique. Par conséquent, un même token peut aboutir à des résultats pour une dimension, et à aucun résultat pour une autre dimension.

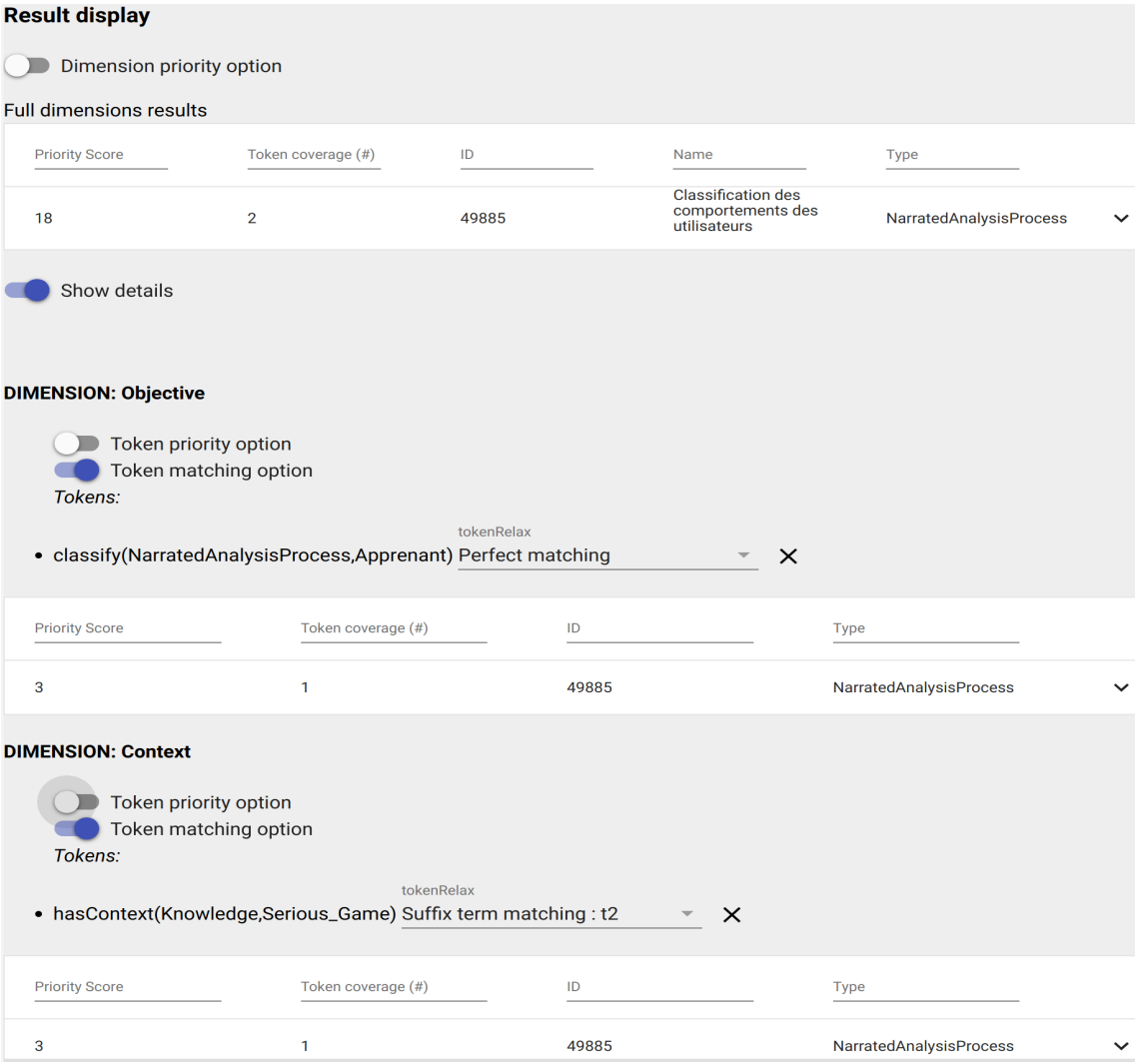


Figure 12.1.: Capture d'écran de **CAPTEN-SEED** : page des résultats potentiels.

requête, et quels éléments du processus ou de l'étape ont permis cela. Les Figure 12.3 et Figure 12.4 montrent ce mécanisme à l'œuvre, respectivement pour le résultat de la dimension de *desiderata* et pour la dimension contextuelle.

Dans ces deux figures, nous voyons que le système indique pour le résultat sélectionné quel token a été trouvé [3]. Puis, pour chaque token, la relaxe utilisée pour mener la recherche est indiquée avant d'indiquer la liste des éléments du résultat qui possèdent le token recherché. Par exemple, dans la Figure 12.4, c'est dans un élément narratif de type contexte, directement lié au processus d'analyse, que le token *Serious_Game* a été identifié ; le tout conditionné par l'utilisation de la relaxe $\mathfrak{Q}_8$. C'est pour cela que le processus a été considéré comme un résultat pertinent : il répond au moins en partie au besoin d'analyse.

Pour représenter cette pertinence vis-à-vis du besoin d'analyse décrit, nous utilisons un score associé à chaque résultat. Ce score est ici associatif. Entendez que si un même élément est identifié plusieurs fois comme résultat dans des dimensions différentes, la valeur finale sera au moins égale à la somme des scores dans chacune des dimensions : les résultats avec le plus haut score sont donc les plus à même de répondre au besoin utilisateur. La raison est que nous voulions permettre à l'utilisateur de pouvoir mettre l'emphase sur certains aspects de son besoin d'analyse.

3. Actuellement, les tokens sont indicés avec des entiers dans le prototype. D'où *The Token (X) has been found :*

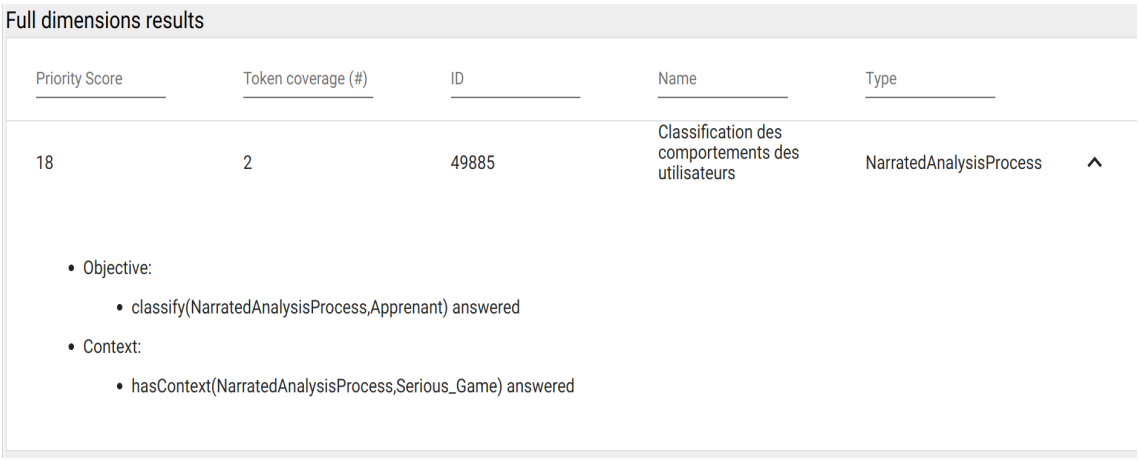


Figure 12.2.: Capture d'écran de **CAPTEN-SEED** : vue détaillée d'un résultat et de sa couverture des dimensions et des tokens.

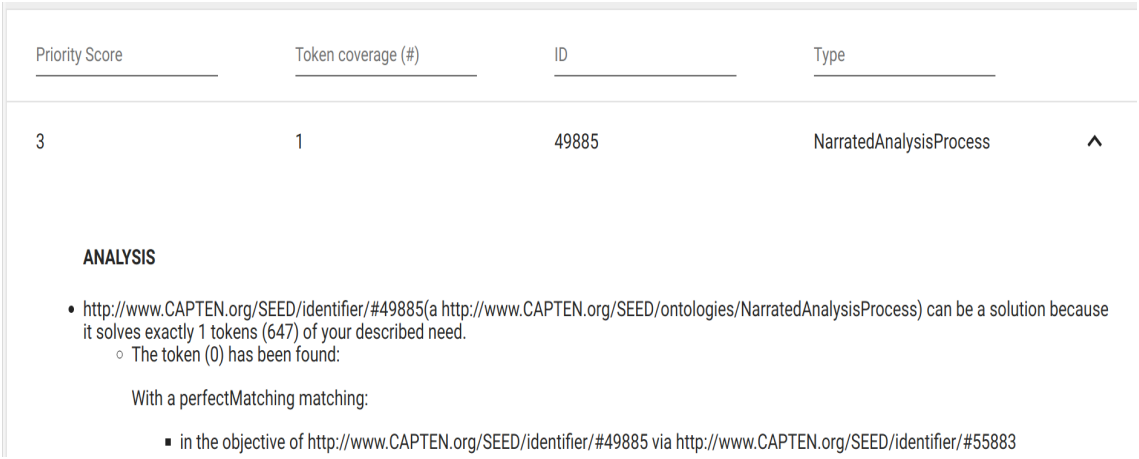


Figure 12.3.: Capture d'écran de **CAPTEN-SEED** : trace de backtrack associée au résultat obtenu dans la dimension de *desiderata*.

Pour cela, nous avons implémenté des modificateurs de priorité à la fois pour les dimensions et pour chacun des tokens qui constituent ces dimensions. Ces modificateurs s'apparentent à des modificateurs flous en cela que nous avons préféré favoriser leur sémantique plutôt que leur précision numérique. Comme le montre la Figure 12.5, nous utilisons des termes comme *High* ou *Normal* pour les représenter. Ensuite, nous nous en servons en tant que coefficient de pondération. Dans le cas d'une dimension, ces coefficients modifient donc directement le score porté par le token associé. C'est le cas dans la Figure 12.5 : comparé à un token sans emphase, le score du 3-uplets *hasContext(Knowledge,Serious_Game)* est plus élevé (4 au lieu de 3).

Enfin, dans le cas de la réponse au besoin d'analyse, ces modificateurs s'appliquent donc à chacune des dimensions. L'utilisateur peut ainsi définir des dimensions comme plus importantes que d'autres lors de sa recherche. Ici, nous favorisons les résultats de la dimension contextuelle en les pondérant avec un coefficient plus important. De cette manière, lorsque plusieurs résultats pertinents sont identifiés, l'utilisateur peut paramétrer la recherche et ainsi les filtrer pour faire ressortir ceux qui sont susceptibles de correspondre au mieux à son besoin d'analyse.

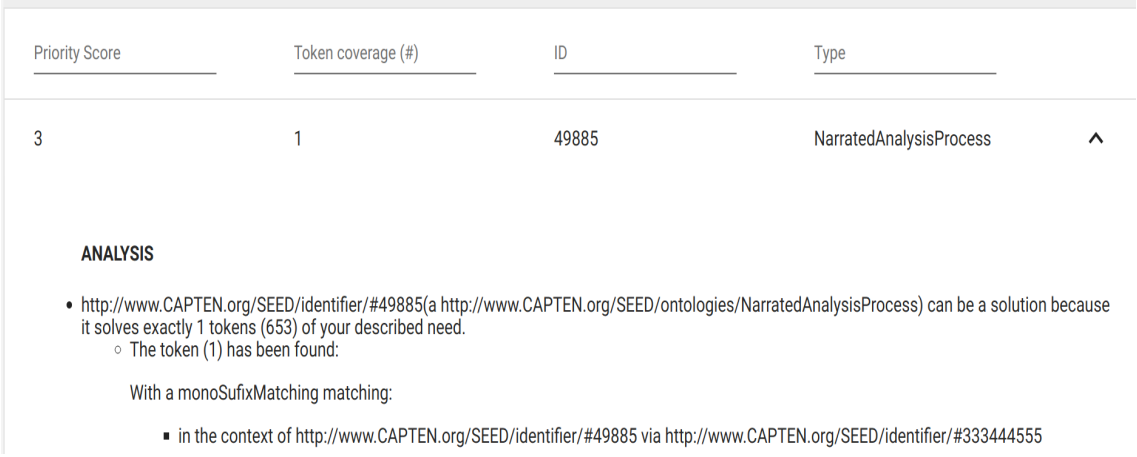


Figure 12.4.: Capture d'écran de **CAPTEN-SEED** : trace de backtrack associée au résultat obtenu dans la dimension contextuelle.

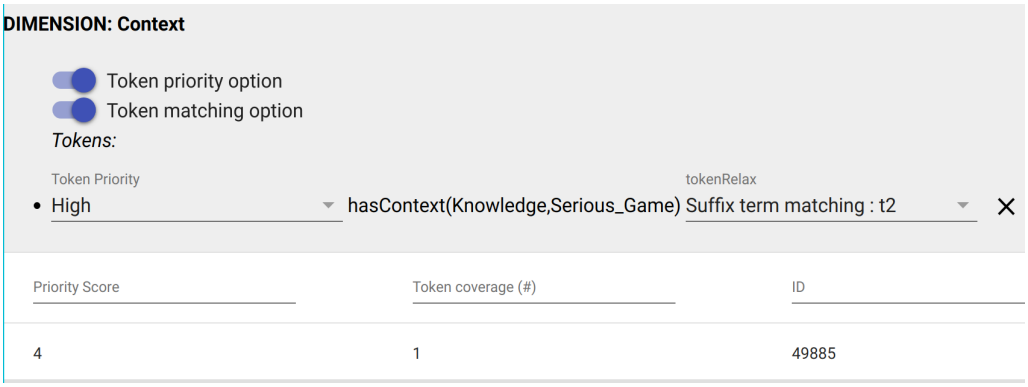


Figure 12.5.: Capture d'écran de **CAPTEN-SEED** : utilisation d'un modificateur flou pour les tokens d'une dimension – ici contextuelle.

12.2 Discussion

CAPTEN-SEED, le prototype que nous avons présenté dans ce chapitre, est un prototype expérimental qui nous a principalement servi de dispositif d'étude et de réflexion sur la portée de la narration dans les mécanismes de recherche et comment donner une place centrale à l'utilisateur. De fait, certains éléments de notre approche **CAPTEN-FRUIT** ne sont pas implémentés, ou pas totalement. Par exemple, nous n'avons pas utilisé de logique floue, mais simplement des coefficients de pondération : il serait intéressant d'étudier la variation des résultats et de leurs scores.

De plus, dans la version présentée du prototype, nous n'avons pas tenu compte des scores d'importance attribuée à chacun des éléments d'un patron de recherche devant être requêtés. La raison principale est que les scores de pertinence deviennent plus complexes à estimer pour un utilisateur. Néanmoins, ces scores d'importance n'en restent pas moins importants, notamment pour d'autres assistances adaptatives (cf. perspectives page 208 et 211). Une solution que nous envisageons est de proposer des scores qui sont normalisés en fonction des éléments du patron les plus contributifs.

En outre, il serait intéressant pour l'utilisateur d'être capable d'exprimer le score global d'un résultat non pas *via* une valeur numérique, mais *via* un terme sémantique. Un tel terme aurait l'avantage de désambiguïser l'évaluation numérique, et de permettre des mises en relation claires avec les dimensions contributives.

Partie IV

Expérimentations et évaluations

Introduction & Plan de la partie

Cette partie concerne la mise à l'épreuve de nos propositions théoriques, présentées dans la Partie II, à travers des expérimentations impliquant principalement des utilisateurs possédant différents niveaux d'expertise d'analyse. L'objectif était de couvrir la pluridisciplinarité inhérente aux différents acteurs impliqués dans le cycle d'élaboration et d'utilisation des analyses que nous avons présenté.

Pour conduire ces expérimentations, nous avons utilisé les prototypes que nous avons présentés dans la Partie III, et qui sont tous disponibles en ligne. Nous présentons donc dans la suite de cette partie la méthodologie adoptée, le retour des évaluations ainsi que les résultats expérimentaux associés.

Ainsi, nous présentons tout d'abord dans le Chapitre 13 la mise à l'épreuve et les résultats de notre proposition **CAPTEN-ALLELE**, visant à émanciper les processus d'analyse de traces des dépendances techniques induites par les outils d'analyse. Nous y observons les effets du changement de paradigme de représentation des analyses sur les différents utilisateurs, et observons aussi la satisfaisabilité des propriétés de répétabilité et de réplicabilité. De plus, nous montrons les raisons qui ont conduit cette approche à servir de matrice à notre proposition narrative.

S'ensuit le Chapitre 14 destiné évaluer notre proposition **CAPTEN-ONION** concernant la narration des processus d'analyse de traces afin de les rendre capitalisables. Nous y évaluons notamment si différents acteurs sont capables d'identifier et de réutiliser des analyses déjà existantes pour répondre à des besoins d'analyse spécifiques. Nous y étudions aussi le rôle de la narration et l'effet de changer - là encore - de paradigme dans la description des processus d'analyse.

Enfin, nous terminons cette partie avec le Chapitre 15. Dans ce chapitre qui concerne **CAPTEN-FRUIT**, l'assistance par la recherche sémantique, nous montrons les résultats expérimentaux que nous avons obtenus lors de nos tests avec les processus d'analyse narrés dans notre prototype. Nous y évaluons notamment la pertinence des résultats produits par notre mécanisme de recherche.

Évaluation de l'abstraction des processus d'analyse de traces

13

Sommaire

Section	**Introduction**	**173**
Section 13.1	**Méthodologie d'évaluation**	**173**
Section 13.2	**Résultats expérimentaux**	**175**
Section 13.3	**Discussion**	**177**

Publications relatives à ce chapitre

(Lebis et al., 2016) A. Lebis, M. Lefevre, V. Luengo et N. Guin (2016). "Towards a Capitalization of Processes Analyzing Learning Interaction Traces". In : *11th European Conference on Technology Enhanced Learning (EC-TEL 2016)*. T. 9891. Lecture Notes in Computer Science. Lyon, France : Springer, p. 397–403

(Lebis, 2016) A. Lebis (2016). "Vers une capitalisation des processus d'analyse de traces". In : *Rencontres Jeunes Chercheurs en EIAH (RJC-EIAH 2016)*. Montpellier, France

Introduction

Ce chapitre concerne l'expérimentation que nous avons menée afin d'évaluer notre proposition théorique pour décrire les processus d'analyse indépendamment des outils d'analyse et des contraintes techniques intégréés à leur description. Afin d'étudier la viabilité initiale de notre approche, cette expérimentation avait deux objectifs principaux. Nous souhaitions d'une part vérifier s'il était possible pour des analystes d'élaborer et de décrire des processus d'analyse en utilisant des concepts plus abstraits – en utilisant ceux de **CAPTEN-ALLELE**, présentés dans le Chapitre 7. Nous voulions d'autre part étudier si notre modèle satisfaisait les différentes propriétés de la capitalisation, et notamment celles de répétabilité et de réplicabilité.

Pour cela, nous avons donc utilisé le prototype que nous avons présenté dans le Chapitre 10, qui nous permet de mettre en œuvre les concepts de variables, d'opérateurs indépendants et de processus d'analyse indépendants. Dans le cadre de ces expérimentations, nous avions défini au sein de ce prototype un ensemble d'opérateurs indépendants (cf. Figure 10.4, page 151) qu'un utilisateur pouvait utiliser pour concevoir son processus d'analyse.

Dans la suite de ce chapitre, nous présentons d'abord le protocole expérimental que nous avons suivi. Ensuite, nous présentons les résultats obtenus lors de cette expérimentation. Enfin, nous discutons ces résultats et expliquons pourquoi cette première proposition s'inscrit en réalité dans un cadre plus important : celui de la narration.

13.1 Méthodologie d'évaluation

Nous avons choisi d'évaluer notre approche qualitativement avec un panel de six personnes possédant des expertises variées : deux informaticiens, trois statisticiens et une experte en sciences cognitives.

Chacune de ces personnes était habituée à travailler dans le domaine des EIAH et avait déjà mené des analyses. Durant cette expérimentation, elles ont été amenées à jouer différents rôles du cycle d'élaboration et d'exploitation de l'analyse (cf. Figure 6.1 de la Section 6.1, page 81).

En effet, ces personnes ont tout d'abord dû remplir le rôle de décideur (R1) puisqu'elles devaient définir et formaliser un besoin d'analyse spécifique, comme nous le verrons après. Elles ont aussi endossé en quelque sorte le rôle de l'expert de l'environnement d'apprentissage (R2), puisqu'elles devaient également décrire les données à utiliser pour réaliser l'analyse. Puis, évidemment, ces personnes sont aussi à considérer comme des analystes (R3), puisqu'elles ont préparé l'analyse et l'ont décrite dans notre prototype.

Concernant le protocole expérimental en lui-même, l'expérimentation a duré 2 heures par personne, et était composée de trois étapes.

La première étape de l'expérimentation consistait à demander au sujet de l'expérimentation de décrire, en dehors du prototype, un besoin d'analyse quelconque, un dispositif d'e-learning auquel un tel besoin s'appliquait, et les traces que ce dispositif pouvait produire. Cette définition était réalisée d'après son expérience personnelle : nous n'avons pas fixé de cadre, ni orienté la description. Le sujet devait ensuite décrire, soit littéralement, soit graphiquement (*e.g.* des croquis d'opérations), un processus d'analyse pouvant répondre à un tel besoin. Nous lui demandions ensuite d'estimer la difficulté du processus ainsi décrit.

Cette partie nous a permis d'étudier comment une analyse pouvait être réfléchie et conçue, alors que l'analyste se trouvait dans un environnement fortement abstrait, et comment son expérience était mobilisée. Il n'avait en effet pas d'outil d'analyse, ni de données concrètes. Lors de cette étape, nous avons observé la granularité utilisée pour décrire les données et les étapes du processus d'analyse. Nous avons aussi étudié la manière dont le sujet agençait les étapes de son analyse, et comment il décrivait les configurations nécessaires. Nous souhaitions aussi savoir si le sujet avait besoin d'utiliser des valeurs concrètes lors de l'élaboration de l'analyse, ce qui n'est pas arrivé avec ces six personnes.

La deuxième étape de l'expérimentation avait pour but de faire découvrir aux sujets notre prototype ainsi que les concepts présentés dans notre proposition théorique. Pour cela, nous les avons guidés, étape par étape, dans la description d'un processus d'analyse indépendant calculant le pourcentage d'étudiants ayant visualisé une vidéo. Cette description a entièrement été réalisée dans notre prototype. Dans cette partie, nous souhaitions principalement savoir si les différents concepts proposés trouvaient un écho chez les analystes et s'ils arrivaient à les manipuler correctement.

Enfin, dans la troisième partie de l'expérimentation, nous avons demandé aux sujets de décrire, en utilisant notre prototype, le processus d'analyse qu'ils avaient décrit sur papier pendant la première partie de l'expérimentation. Cela nous a permis d'étudier s'il était possible de décrire avec notre prototype des processus d'analyse variés. Nous souhaitions également identifier d'éventuelles divergences conceptuelles entre les processus décrits sur papier et ceux décrits avec notre prototype. De plus, cette étape nous a permis d'étudier le comportement des analystes lorsqu'ils étaient confrontés à l'utilisation de concepts plus abstraits que les objets qu'ils sont accoutumés à manipuler dans les outils d'analyse dont ils ont l'habitude.

Tous les sujets ont été autonomes durant ces expérimentations, même si nous avons répondu à leurs questions. Ils ont été filmés durant cette expérimentation et nous avons aussi rapporté les observations conduites pendant leur utilisation du prototype dans une grille d'évaluation qualitative portant principalement sur la faculté des analystes à comprendre et à utiliser les différents concepts proposés. Les sujets ont en outre répondu à un questionnaire après chacune des parties 2 et 3. Le prototype, ainsi que tous les documents et productions liés à ces expérimentations et diffusables, sont accessibles en ligne (Lebis, 2018[f]) et en annexe, Section D, page 239.

13.2 Résultats expérimentaux

Le premier résultat notable de cette expérimentation est que tous les sujets ont pu réaliser, avec notre prototype, un processus d'analyse indépendant répondant au besoin d'analyse qu'ils avaient choisi dans la première partie de l'expérimentation. Deux processus n'ont cependant pas pu être achevés, l'un pour cause de manque de temps, l'autre parce qu'il manquait dans notre prototype la description de certains opérateurs indépendants qui auraient été nécessaires. Malgré cela, ce premier résultat indique que notre approche permet de répondre aux propriétés de réplicabilité et de répétabilité. En effet, la description de la succession des opérations semble s'opérer correctement avec des concepts plus abstraits. De plus, les informations nécessaires pour répéter l'analyse sont présentes *via* une description des variables initiales nécessaires, la configuration des opérateurs indépendants, et les variables terminales du processus.

		Compréhension	Utilisation	Observation
Variable		**Compris** : il ne s'agit pas de la valeur de la donnée	Évident à manipuler	**Manque** de relations entre les variables
Liste		**Confusion** : régulièrement assimilée à des variables particulières, et pas comme des structures organisatrices	Des **difficultés** à accéder à des variables précises dans l'enchaînement d'opérations	**Manque** de visibilité, d'informations et de sémantiques. Manque de relations, notamment avec les variables
Opérateur indépendant		**Compris** : opération abstraite, agissant comme une boîte noire conceptuelle	Facile à appliquer sur les variables pour produire des résultats	**Couverture** : on ne sait pas s'il est possible de représenter tous les opérateurs
	Entrée	**Compris** : les variables sont prises en compte par l'opération	Certaines **difficultés** pour choisir les variables appropriées	**Manque** de flexibilité
	Sortie	**Compris** : les variables représentent l'état courant après application	Utilisation facilitée grâce à la génération assistée par nos modèles	**Besoin** de feedbacks
	Paramètres	**Compris** : concepts intervenant dans la configuration de l'opérateur	**Compliqué** à utiliser par manque de sémantique	**Manque** de sémantiques et d'informations
Processus d'analyse indépendant		**Compris** : décrit une analyse à un niveau abstrait, plus proche de la méthodologie d'analyse	Des **difficultés** à enchaîner les opérateurs de manière pertinente	**Besoin** de plus de sémantique, d'informations, de flexibilité et d'opérateurs indépendants

Tableau 13.1.: Synthèse des grilles d'observations et des questionnaires de l'évaluation de **CAPTEN-MANTA**.

De plus, tous les sujets semblent avoir correctement compris notre proposition et le rôle de chacun des concepts introduits, comme le montrent les retours utilisateurs et nos observations reportées dans la première colonne du Tableau 13.1. En effet, les analystes étaient conscients que le processus d'analyse qu'ils ont décrit avec le prototype était abstrait et indépendant des outils d'analyse, et qu'ils manipulaient des concepts d'opérations, représentés par les opérateurs indépendants. Ils ont également assimilé le fait que les variables représentent les concepts véhiculés par les traces et que les variables initiales servent à contraindre l'utilisation des processus d'analyse indépendants. L'un des analystes a, par exemple, remarqué que les variables initiales sont "[...] strictement le sous-ensemble essentiel pour faire un processus d'analyse".

Cependant, les résultats expérimentaux montrent plusieurs limitations importantes concernant notre proposition et soulèvent de nouvelles questions (cf. troisième colonne du Tableau 13.1). Premièrement,

nous avons remarqué que l'élaboration de certains processus d'analyse a été, à certains moments, entravée par un manque d'opérateurs indépendants. Cela nous fait nous interroger sur la manière d'obtenir un sous-ensemble suffisant d'opérateurs, afin que notre proposition soit utilisable pour une grande variété d'analyses, tout en s'assurant que la sémantique de ces opérateurs soit conservée.

Nous avons également noté un manque important de rétroactions du fait que les processus d'analyse indépendants ne sont pas exécutables, malgré la possibilité offerte par notre prototype d'identifier quels outils d'analyses sont capables d'implémenter ces processus d'analyse indépendants. D'après les retours utilisateurs, des informations concernant l'évolution des variables et de leurs valeurs étaient attendues, principalement pour savoir si les choix de paramétrage des opérateurs indépendants étaient corrects, et si les opérateurs indépendants eux-mêmes étaient pertinents. En effet, comme le laisse suggérer par exemple Fayyad & *al.* (FAYYAD et al., 1996), l'analyse est une tâche itérative et il est courant de raffiner le paramétrage d'un opérateur en vérifiant les valeurs obtenues. Notre approche narrative (cf. Chapitre 8) permet de renforcer ce point.

En outre, l'expérimentation a également mis en avant le fait que le paramétrage des opérateurs indépendants, tel qu'il est défini dans notre modèle, doit être plus précis que simplement indiquer le type de paramétrage attendu : il faut permettre de représenter les effets du paramétrage sur l'opérateur. Il est arrivé que les paramètres aient mal été interprétés par les sujets de l'expérimentation (*e.g.* ambiguïté dans les configurations d'un opérateur de filtre), et que ces personnes n'aient pas toujours su comment renseigner les paramètres, ces derniers s'exprimant sous la forme de texte libre. Ces résultats expérimentaux montrent ainsi que notre approche ne répond pas entièrement à la propriété de réutilisabilité, puisqu'il semble important de pouvoir représenter l'effet des opérations sur les variables, ainsi que de les expliquer, et ce d'une manière structurée mais toujours indépendante des outils d'analyse.

De plus, nous avons également remarqué que notre modèle ne permettait pas aux sujets de décrire toutes les informations qu'ils auraient aimé donner sur les processus d'analyse qu'ils ont décrits. On peut citer le fait d'expliciter la signification de certaines variables des traces, le fait de comprendre la signification d'un paramètre spécifique pour un opérateur donné, ou encore expliquer la raison pour laquelle une étape est réalisée. D'après eux, cela leur a posé des difficultés pour réaliser certaines étapes du processus indépendant qu'ils devaient décrire lors de la troisième partie de l'expérimentation.

Aussi, les listes pensées comme des conteneurs prévus pour structurer et sémantiser les variables n'ont pas fonctionné ; dans certains cas, elles ont même eu l'effet inverse en complexifiant la description. D'après les retours d'expérimentations, cela est dû au manque d'informations explicites pouvant exister entre les variables d'une même liste, et les listes automatiquement créées par les opérateurs indépendants.

De surcroît, nous avons comparé les analyses décrites dans notre prototype et celles sur papier lors de la première partie de l'expérimentation, en cherchant les différences apparentes (*e.g.* nombre d'étapes différentes, paramétrages différents, variables différentes ou encore noms différents). Par exemple, nous avons ainsi constaté qu'une analyse décrite dans notre prototype ne proposait pas de classer exactement de la même manière que celle sur papier les étudiants – notamment en n'utilisant pas les mêmes classes ; nous avons aussi remarqué la fusion de deux étapes dans une autre analyse – changeant également la signification de score de la variable de sortie en ratio. Nous avons donc constaté de légères divergences sémantiques entre certaines étapes des analyses décrites sur papier dans la première partie de l'expérimentation et celles décrites avec le prototype dans la troisième partie. Ce constat est intéressant, puisqu'il impose de nous interroger sur la raison d'une telle divergence. Il peut en effet s'agir d'une éventuelle ingérence technique de notre outil dans la description du processus d'analyse indépendant, ou d'un manque de description de la sémantique portée par les opérateurs indépendants intégrés dans le prototype.

Enfin, notons que notre proposition théorique a été bien reçue par les sujets. Ils l'ont en effet jugée pertinente, puisque l'abstraction réalisée leur a permis de "formaliser le processus à implémenter venant d'autres outils d'analyses" et les a incités à "organiser les différentes idées".

13.3 Discussion

Ainsi, notre proposition de représenter des processus d'analyse indépendamment des outils d'analyse nous permet de répondre principalement aux propriétés de réplicabilité et de répétabilité, socle de la capitalisation (cf. Figure 6.2, page 88). En effet, les processus ainsi créés sont identifiables, au sens où chaque opération est représentée de manière non ambigüe et peut donc être mise en œuvre dans des outils d'analyse [1], et les entrées, paramétrages et sorties attendus sont aussi représentés.

Il est également possible de considérer que l'ouverture de ces processus est renforcée, grâce à l'utilisation d'une modélisation qui regroupe les opérations issues de différentes sources. Cependant, le manque de sémantique et d'informations associées aux différents éléments de l'analyse empêche sa compréhension, et limite drastiquement les possibilités de réutilisation de tels processus d'analyse indépendants, ne serait-ce qu'avec l'apparition d'une divergence sémantique entre processus d'analyse implémentés et processus d'analyse indépendants.

En effet, comme nous l'avons vu, les contraintes techniques ne peuvent pas être considérées comme la seule limitation à la capitalisation des processus d'analyse de traces d'apprentissage. En observant les analyses décrites par les sujets de l'expérimentation – à la fois sur papier et dans notre prototype, l'on voit que la situation pédagogique dans laquelle elles ont été conduites est intégrée de manière implicite au processus d'analyse implémenté (*e.g.* présupposé sur le contenu des traces, sur les actions possibles des apprenants). Cependant, il nous est impossible d'en tenir compte. Or, comme nous l'avons montré durant l'état de l'art, ce contexte pédagogique pose des difficultés pour réutiliser et adapter un processus d'analyse dans un autre contexte. De plus, ces contraintes contextuelles ne sont actuellement pas prises en compte dans les travaux menés au sein de la communauté EIAH sur l'ouverture des processus d'analyse, et seulement quelques travaux dans d'autres disciplines, comme ceux des Research Objects (Belhajjame et al., 2013a), tentent de le décrire.

En croisant nos résultats expérimentaux et ces travaux issus d'autres disciplines, l'on a remarqué qu'il était nécessaire d'ajouter à notre approche d'indépendance technique une représentation sémantique du contexte pédagogique. Toutefois, il est tout aussi important d'être en mesure de représenter l'information d'une manière générale, et de l'intégrer au processus d'analyse pour le rendre compréhensible par lui-même. Nous avons aussi constaté qu'il était important d'étendre le feedback des différentes opérations appliquées sur les variables pour renforcer la compréhension et donner des perspectives d'adaptation aux différents acteurs.

En outre, sachant qu'une approche ontologique peut également bénéficier à la description du processus d'analyse lui-même, elle permet alors de s'approcher d'une représentation unifiée des processus d'analyse et ainsi favoriser la manière de les exploiter : c'est la raison pour laquelle nos travaux suivants (*i.e.* **CAPTEN-ATOM** et **CAPTEN-AERIS**) se sont établis dans un cadre ontologique. L'ontologie sous-jacente à **CAPTEN-ATOM** permet de structurer l'ensemble des informations qui permettent à un utilisateur de comprendre et de réutiliser un processus d'analyse.

1. Notre prototype indiquant, pour chaque étape, quels outils d'analyse peuvent être utilisés

Évaluation de la narration des processus d'analyse de traces

14

Sommaire

Section	**Introduction**	**179**
Section 14.1	**Méthodologie d'évaluation**	**180**
Section 14.2	**Résultats expérimentaux**	**181**
14.2.1	Résultats généraux	182
14.2.2	Effet sur la compréhension	183
14.2.3	Effet sur l'adaptation	184
14.2.4	Effet sur la réutilisation	184
14.2.5	Effet sur la capitalisation en général	185
Section 14.3	**Discussion**	**186**

Publications relatives à ce chapitre

(Lebis et al., 2017a) A. Lebis, M. Lefevre, V. Luengo et N. Guin (2017a). "Approche narrative des processus d'analyses de traces d'apprentissage : un framework ontologique pour la capitalisation". In : *Environnements Informatiques pour l'Apprentissage Humain*. EIAH 2017. Strasbourg, France

(Lebis et al., 2018a) A. Lebis, M. Lefevre, V. Luengo et N. Guin (2018a). "Capitalisation of Analysis Processes : Enabling Reproducibility, Openess and Adaptability thanks to Narration". In : *LAK '18 - 8th International Conference on Learning Analytics and Knowledge*. Sydney, Australia : ACM, p. 245–254

Introduction

Ce chapitre concerne l'expérimentation que nous avons menée afin d'évaluer notre approche narrative des processus d'analyse de traces pour permettre, *in fine*, leur capitalisation. Or, comme nous l'avons vu, la capitalisation se révèle être une propriété composite. Par conséquent, dans cette expérimentation, nous avions plusieurs aspects à observer. Premièrement, nous souhaitions étudier s'il était possible de décrire et de représenter les processus d'analyse, les informations et les choix associés, et les relations existant entre tous ces éléments, le tout avec des contraintes sémantiques fortes et des concepts de haut niveau.

Deuxièmement, nous voulions étudier l'effet de notre proposition sur les différentes propriétés de la capitalisation, notamment celles que nous n'avions pas pu couvrir avec notre approche concernant l'indépendance technique : la compréhension, la réutilisation et l'adaptation des processus d'analyse. Cela nécessite alors un dispositif expérimental complexe où des analyses ont déjà été narrées et où différents acteurs vont être en mesure de les consulter pour pouvoir répondre à leur propre besoin d'analyse. La Figure 14.1 ci-contre illustre les différents comportements des acteurs pour réutiliser les analyses lorsque la capitalisation est effective. Il s'agit d'une vue plus précise de la Figure 6.4, page 90, à propos du cycle d'élaboration et d'exploitation de l'analyse, qui place le processus d'analyse au centre de ces comportements et qui illustre les interactions qui surviennent lors de la capitalisation.

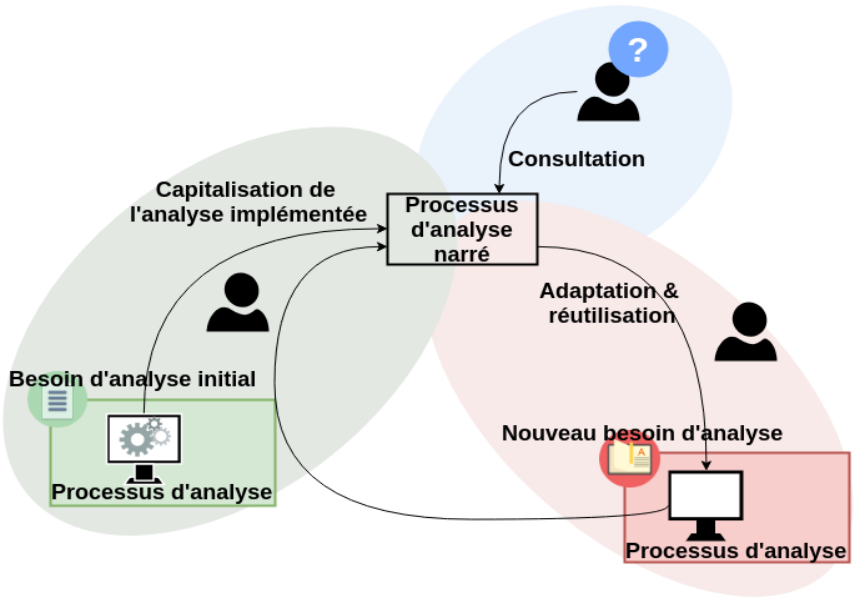


Figure 14.1.: Vue détaillant l'effet et l'apport de notre approche narrative pour la capitalisation des processus d'analyse.

Aussi, nous pensons que pour évaluer correctement l'efficacité de notre approche en ce qui concerne la capitalisation, il est important de soumettre les sujets d'expérimentation à une phase de consultation et de compréhension de l'existant : la partie bleue (en haut à droite) de la Figure 14.1. Puis, de les soumettre à une phase d'adaptation des éléments existants jugés pertinents pour répondre à leurs besoins, afin de les réutiliser ensuite. Cela constitue alors la partie rouge (en bas à droite) de la Figure 14.1 – avec éventuellement un enrichissement de l'existant.

Pour définir cet existant, nous avons, préalablement à l'expérimentation et dans notre prototype **CAPTEN-TORTOISE**, narré des processus d'analyse déjà implémentés. Ce faisant, nous avons ainsi joué le rôle de l'acteur présent dans la partie verte (en bas à gauche) de la Figure 14.1, et nous avons pu évaluer nous-même si les propriétés de réplicabilité et de répétabilité étaient bien conservées entre notre proposition précédente et cette proposition narrative [1].

Dans la suite de ce chapitre, nous présentons d'abord le protocole expérimental que nous avons suivi. Ensuite, nous présentons les résultats obtenus lors de cette expérimentation. Enfin, nous discutons ces résultats.

14.1 Méthodologie d'évaluation

Nous avons évalué notre approche avec un panel de six personnes [2]. Ces personnes avaient toutes l'habitude de travailler dans le domaine des EIAH et des Learning Analytics. Chaque séance, impliquant une seule personne à la fois, a duré trois heures. Nous leur avons demandé d'évaluer leur expertise sur l'analyse de traces, sur une échelle de zéro à dix, zéro signifiant "aucune expertise" et dix "expert". Dans l'ensemble, ces personnes étaient habituées à mener des analyses, avec une moyenne de 5.8 $(3, 5, 6, 6, 7, 8)$ [3]. Les résultats montrent qu'il s'agit d'une population hétérogène dans l'analyse de traces, nous permettant ainsi de tester notre approche avec des profils variés.

La première partie de l'expérimentation consistait à présenter au sujet de l'expérimentation notre approche narrative. Nous lui montrions le fonctionnement de notre prototype **CAPTEN-TORTOISE**,

1. Nous avons présenté ces narrations dans le prototype **CAPTEN-TORTOISE** dans la Section 11.3, page 159, de la Partie III.
2. Cet ensemble de personnes est différent de celui de l'expérimentation précédente présentée en section 13
3. Les nombres entre parenthèses représentent le score de chacune des personnes.

ainsi que certains processus d'analyse narrés qui y étaient définis, tout en les expliquant. Lors de ces explications, nous portions un point d'intérêt particulier aux éléments narratifs utilisés, sur la présence des relations que cela entraînait, ainsi qu'à l'ontologie et au vocabulaire contrôlé permettant de choisir les valeurs de ces éléments narratifs. Dans l'ensemble, cette première partie durait trente minutes.

Dans la deuxième partie de l'expérimentation, la personne devait mettre en œuvre une analyse pour répondre à l'un de ces deux besoins d'analyse : "Prédire la certification d'un étudiant à la fin d'un cours" ou "Identifier les types d'apprenants, et si possible, par cours". L'analyste pouvait choisir parmi les deux besoins qui lui étaient proposés celui qu'il/elle préférait. Nous fournissions, pour chacun de ces deux besoins, des traces de MOOC (MITx et HarvardX, 2014) sur lesquelles devait porter l'analyse, ainsi que leur documentation.

L'objectif de cette deuxième partie était d'une part d'évaluer les compétences d'analyse effectives des sujets. D'autre part, cela nous donnait la possibilité d'étudier concrètement leurs habitudes d'analyse : autrement dit si la personne se documentait avant de réaliser l'analyse, si elle consultait l'existant, et si oui, comment elle y accédait et comment elle le réutilisait, etc.

Pendant une heure et demie, la personne était donc autonome et était autorisée à accéder à n'importe quelles ressources susceptibles de l'aider dans sa tâche d'analyse, excepté notre prototype. De plus, la personne pouvait choisir le ou les outils d'analyse qu'elle souhaitait utiliser. À la fin du temps alloué, que la personne ait terminé ou non son analyse, nous lui donnions accès à **CAPTEN-TORTOISE**, pour la troisième partie de l'expérimentation.

Dans cette troisième partie, nous proposions aux personnes d'utiliser notre prototype pour consulter un ensemble de processus d'analyse narrés (cf. Section 11.3, page 159). Elles pouvaient alors s'en inspirer si besoin afin de terminer ou d'améliorer leur analyse. Pendant une heure, chaque personne était laissée en autonomie autant que possible, bien que nous ne nous étions pas interdit d'intervenir pour expliquer certaines interfaces du prototype.

Enfin, les quinze dernières minutes étaient consacrées à répondre à un formulaire. Ce formulaire interrogeait les sujets sur le déroulement de l'expérimentation, sur les difficultés rencontrées, et sur leur point de vue concernant le prototype en lui-même. Il recensait également l'opinion de l'analyste concernant notre proposition, ses modèles sous-jacents, et sa pertinence pour la capitalisation des processus d'analyse de traces d'apprentissage au sein de la communauté. Le prototype utilisé durant les expérimentations, ainsi que le protocole, sont accessibles en ligne (Lebis, 2018[d]) et en annexe, Chapitre E, page 259.

14.2 Résultats expérimentaux

À la fin de la deuxième partie de l'expérimentation (soit avant l'utilisation du prototype par le sujet de l'expérimentation), deux sujets avaient fini le processus d'analyse répondant au besoin qu'ils avaient choisi. Nous leur avons ensuite demandé d'estimer la qualité de leur analyse en utilisant une échelle numérique allant de 0 à 5, signifiant respectivement *Insatisfaisant* à *Totalement satisfaisant*. Ces deux personnes l'ont évalué à 3. Trois autres analystes ont rencontré des difficultés lors de l'élaboration de l'analyse. Enfin, bien qu'ayant des premiers résultats, un dernier analyste n'a pas pu finir par manque de temps et a estimé la qualité de son analyse à 1.

Dans la troisième partie de l'expérimentation, l'intérêt d'utiliser notre prototype dépendait directement, pour la personne, de son avancement dans la mise en œuvre de l'analyse. Lorsqu'elle avait terminé la mise en œuvre de l'analyse, il s'agissait alors d'utiliser le prototype pour explorer des processus d'analyse existants. En effet, cela offrait une base propice à l'amélioration de la qualité globale de l'analyse implémentée, en comparant son approche avec celles des analyses décrites dans notre prototype et en identifiant d'autres techniques de résolution permettant de répondre au besoin d'analyse.

Lorsque le sujet n'avait pas pu terminer son analyse, il s'agissait alors d'utiliser le prototype pour tenter de la terminer. Pour ce faire, l'analyste devait chercher au sein du prototype les processus d'analyse

narrés, ou une partie de ces processus, pouvant répondre à son besoin, en consultant les éléments narratifs ainsi que la structuration en étapes du processus d'analyse narré. Puis, il essayait, avec l'aide de ces éléments narratifs, d'adapter les processus d'analyse qu'il avait jugés pertinents, puis de les implémenter dans son outil d'analyse.

De cette manière, seulement une seule personne, à la fin de l'expérimentation, n'a pas terminé la mise en œuvre de son analyse, même avec notre prototype. La raison ici est un manque de temps. Une autre personne n'a, quant à elle, pas eu besoin d'utiliser notre prototype pour améliorer son analyse, comme nous le verrons plus loin. C'est la raison pour laquelle certains résultats ne concernent que cinq sujets d'expérimentation, et non pas les six.

14.2.1 Résultats généraux

En guise de première observation générale de l'expérimentation, nous nous intéressons à la répartition des choix des besoins d'analyse. Quatre sujets d'expérimentation ont choisi le premier besoin (*i.e.* Prédire la certification d'un étudiant à la fin d'un cours), et les deux autres le deuxième besoin (*i.e.* Identifier les types d'apprenants, et si possible, par cours). Cette répartition est expliquée par les sujets eux-mêmes dans le questionnaire, précisant qu'ils ont choisi celui où ils avaient la meilleure intuition sur la manière de le résoudre. C'est une observation intéressante à la fois pour l'expérimentation et pour les résultats. En effet, cela nous permet de savoir, à l'instar de l'expérimentation précédente (cf. Chapitre 13), que ces personnes ont été capables de mobiliser et de réutiliser des connaissances d'analyse précédemment acquises – qui pourront potentiellement servir lors de la consultation des processus narrés.

Ensuite, et puisque nous n'imposions pas d'outils d'analyse particuliers, les analystes ont eu recours à une diversité d'entre eux pour conduire leurs analyses : Excel, RapidMiner, R, Coheris Analytics SPAD and SAS Enterprise Miner. Cela vient étayer le constat présenté dans notre état de l'art à propos de la diversité des outils d'analyse et de la diversité de modélisation des opérations disponibles. Nous nous sommes intéressés aux critères principaux conditionnant le choix des outils en question. Ils se révèlent être, pour quatre des six personnes, liés à leur expertise quant à leur utilisation. Toutefois, pour trois des six personnes, l'efficacité de l'outil d'analyse est également un critère de choix. Par ailleurs, il faut noter que deux personnes ont eu recours à deux outils d'analyse différents.

Nous avons aussi voulu évaluer l'effet de notre prototype **CAPTEN-TORTOISE** sur l'analyse que les personnes avaient ou étaient en train de mettre en œuvre. À la question "CAPTEN a-t-il été bénéfique pour votre analyse ?", une personne a répondu que "Non, CAPTEN ne m'a rien apporté de plus". La raison était qu'il avait déjà mené des analyses très similaires sur des traces analogues. Il n'a ainsi pas eu recours à notre prototype pour améliorer son processus d'analyse. Toutefois, il s'est tout de même intéressé à l'existant narré et s'y est comparé.

À cette même question, les cinq autres analystes ont répondu que "Oui, j'ai pu améliorer mon analyse" grâce au prototype. Quatre des cinq analystes ont indiqué qu'ils ont réutilisé des processus d'analyse narrés, ou parfois des étapes, dans leur propre processus d'analyse, dans le but de l'améliorer. Le dernier analyste a indiqué qu'utiliser CAPTEN lui a permis d'avoir connaissance d'autres méthodes d'analyse et lui a permis de finaliser son processus d'analyse. **CAPTEN-TORTOISE** a également été utilisé pour rechercher des informations précises par quatre de ces cinq personnes. Certaines de ces informations concernaient le choix des variables appropriées pour réaliser l'analyse. D'autres informations concernaient le choix des opérateurs à utiliser : la justification de la pertinence de l'opérateur, la configuration effectuée, et l'étape dans laquelle l'utiliser au sein du processus.

Nous avons ensuite voulu évaluer le point de vue des sujets quant à la qualité des analyses obtenues suite à l'utilisation de notre prototype, à la fin de la troisième partie de l'expérimentation. Nous avons obtenu une moyenne de 3.5 ($3; 3; 4; 4$ [4]) avec une échelle allant de 0 à 5, soit une différence positive de 1.2 entre des analyses conçues sans utiliser **CAPTEN-TORTOISE** (cf. introduction de la section) et celles conçues avec l'aide de ce prototype. Tout en gardant en considération que la qualité de l'analyse

4. Le sujet d'expérimentation n'ayant pas terminé son analyse et celui n'ayant pas utilisé le prototype pour améliorer son analyse n'ont pas répondu à cette question.

Éléments de CAPTEN-ATOM	Trace (sur 2 sujets)	Contexte (sur 3 sujets)	Analyse narrée (sur 5 sujets)
Processus d'analyse narré	1 / 2	3 / 3	5 / 5
Opérateur narré	1 / 2	0 / 3	3 / 5
Graphe de variables	1 / 2	1 / 3	4 / 5
Étape	2 / 2	3 / 3	5 / 5
Connaissance	0 / 2	0 / 3	4 / 5
Élément narratif	0 / 2	2 / 3	5 / 5

Tableau 14.1.: Résultats concernant l'effet des différents éléments de notre approche narrative sur la compréhension des traces et du contexte du nouveau besoin, et des processus d'analyse narrés. Le nombre dans chaque cellule indique le nombre de sujets ayant répondu positivement sur le nombre de sujets concernés. Ce ratio conditionne la couleur de la case : lorsqu'il est inférieur ou égal au tiers, la case est rouge ; lorsqu'il est supérieur ou égal aux deux tiers, la case est verte ; orange autrement .

a pu être améliorée grâce au temps supplémentaire alloué par la troisième partie de l'expérimentation, ces résultats restent très encourageants sur l'apport de notre approche.

14.2.2 Effet sur la compréhension

Dans cette section, nous nous intéressons à l'effet de notre approche narrative sur la compréhension des analyses narrées. Autrement dit, comment les différents éléments de notre approche contribuent à identifier si une analyse peut se révéler utile pour répondre au besoin d'analyse d'une personne. Nous nous intéressons aussi à un éventuel effet sur la compréhension des traces et du contexte de l'analyse relatif au besoin d'analyse auquel il est nécessaire de répondre. L'objectif ici est d'identifier si notre approche narrative, en plus de fournir une meilleure compréhension de l'existant, permet de mieux comprendre de nouveaux besoins d'analyse.

Pour évaluer la compréhension des traces à disposition pour répondre à leur besoin d'analyse, nous avons demandé aux analystes d'indiquer leur compréhension initiale des traces. L'échelle de notation utilisée allait de 0 à 5, signifiant respectivement *Incomprises* à *Totalement comprises*. Il apparaît que ces traces semblent avoir été comprises dans l'ensemble $(3; 3; 4; 4; 5; 5$, avec une moyenne de $4)$. Nous avons procédé de la même manière pour la compréhension initiale du contexte de l'analyse. En revanche, leur compréhension se relève un peu plus mitigée $(1; 3; 4; 4; 5; 5$, avec une moyenne de $3.6)$.

Nous avons ensuite cherché à savoir si notre approche était également capable d'apporter une assistance à la compréhension des traces et du contexte d'un besoin d'analyse. Pour cela, nous le leur avons demandé. Dans le cas d'une réponse positive, nous leur avons ensuite demandé quels éléments de notre approche avaient contribué à l'amélioration de la compréhension. Il s'avère que deux analystes ont indiqué que l'utilisation de notre prototype a permis d'améliorer leur compréhension des traces fournies, grâce à la consultation de plusieurs processus d'analyse narrés. Les éléments de notre approche qui ont été identifiés par ces deux personnes comme utiles à l'amélioration de la compréhension des traces initiales sont présentés dans la colonne *Trace* du Tableau 14.1. Ici, bien que l'on puisse s'attendre à ce que les graphes de variables aient pu assister ces deux sujets, c'est principalement des étapes de processus d'analyse narrés qui ont été utiles pour renforcer la compréhension des traces.

Trois analystes ont également indiqué que leur compréhension du contexte de l'analyse a été améliorée grâce à **CAPTEN-TORTOISE**. La colonne *Contexte* du Tableau 14.1 montre, pour ces trois analystes, quels éléments de notre approche ont été impliqués. On constate ainsi que c'est à la fois les éléments narratifs et la structuration du processus en étapes qui les a aidés à mieux comprendre le contexte.

Enfin, nous avons demandé aux cinq analystes ayant utilisé notre prototype d'indiquer quels éléments les ont aidés pour comprendre les processus d'analyse narrés. Les résultats sont présentés dans la dernière colonne du Tableau 14.1. Nous y voyons ainsi que c'est la combinaison de la structuration en étapes et des éléments narratifs qui permet de satisfaire la propriété de compréhension des processus d'analyse, nécessaire à leur capitalisation.

Notons tout de même qu'il n'a pas toujours été aisé pour les utilisateurs de trouver des informations au sein de notre prototype. En effet, en interrogeant les personnes de l'expérimentation avec une échelle de notation allant de 0 à 5, signifiant respectivement *Trivial* et *Complexe*, à la question "quelle a été la difficulté pour trouver des informations qui vous ont aidé", nous avons obtenu les notes $(1; 2; 3; 4; 4)$.

14.2.3 Effet sur l'adaptation

Dans cette section, nous rapportons l'effet de notre approche sur l'adaptation de l'existant. En effet, afin de terminer ou d'améliorer la réponse à leur besoin d'analyse par la mise en œuvre d'une analyse, les sujets pouvaient utiliser l'existant décrit dans notre prototype. Comme nous l'avons vu, ce fut le cas de cinq des sujets. Toutefois, avant de réutiliser l'existant, ils devaient d'abord logiquement identifier quels processus et étapes étaient pertinents – principalement en comprenant les aspects de ces éléments, puis ils devaient être en mesure de les adapter.

Aussi, nous avons évalué la réussite des sujets à adapter les processus d'analyse narrés ou certaines étapes de ces processus existants qui semblaient selon eux répondre à leur besoin. Pour cela, nous les avons interrogés en utilisant une échelle de notation allant de 0 à 5, signifiant respectivement *Absolument pas* et *Totalement*. Les résultats sont présentés dans la Figure 14.2. Ces résultats préliminaires sont extrêmement motivants, sachant que l'adaptation d'un processus d'analyse est une tâche complexe, comme nous l'avons abordé lors de l'état de l'art.

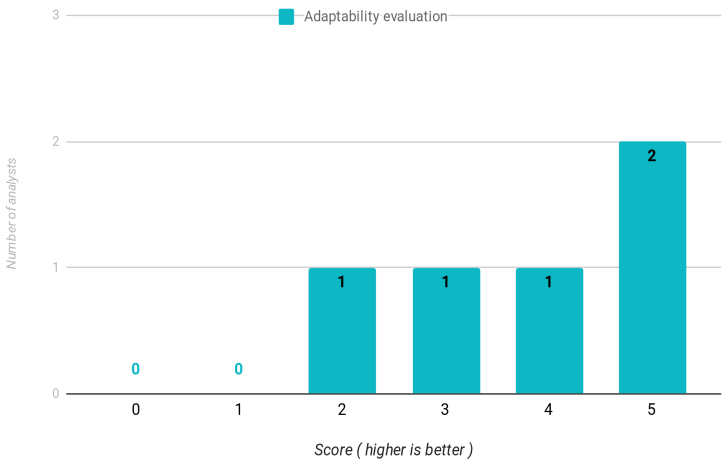


Figure 14.2.: Résultats concernant la faculté des analystes à adapter les processus d'analyse narrés choisis, ou certaines de leurs étapes, d'après une échelle allant de 0 à 5, signifiant respectivement *Absolument pas* et *Totalement*.

Nous nous sommes ensuite intéressés à la place de notre prototype, en tant qu'assistance, dans l'adaptation des processus d'analyse narrés et des étapes. Pour cela, nous avons utilisé une échelle de notation allant de 0 à 10, signifiant respectivement *Inutile* et *Indispensable*. Les résultats, visibles dans la Figure 14.3, montrent que **CAPTEN-TORTOISE** fournit une bonne assistance à l'adaptation des processus d'analyse (avec une moyenne égale à 6). Cela est d'autant plus encourageant que notre prototype n'implémente qu'un sous-ensemble de notre ontologie (cf. Chapitre 11, page 153).

14.2.4 Effet sur la réutilisation

Dans cette section, nous rapportons le degré de réussite des sujets concernant la réutilisation par l'implémentation des processus d'analyse et des étapes préalablement adaptés. Avant cela, nous étudions aussi, pour les sujets ayant réutilisé des éléments, le degré de similarité estimé entre les informations disponibles pour élaborer l'analyse (traces, description du besoin, et contexte) et celles présentes dans les processus narrés. L'objectif était de voir si notre approche permet de réutiliser des processus qui ne sont pas seulement identiques, du point de vue de l'utilisateur.

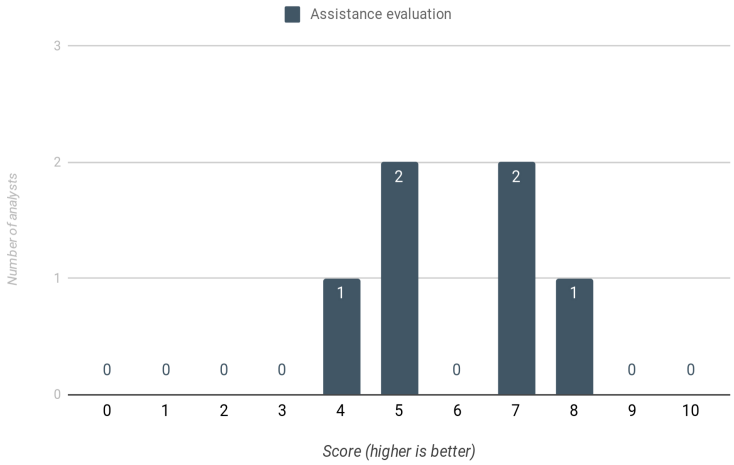


Figure 14.3.: Résultats concernant l'assistance fournie par CAPTEN, d'après une échelle allant de 0 à 10, signifiant respectivement *Inutile* et *Indispensable*.

Pour ce faire, nous avons demandé aux cinq analystes d'estimer le niveau de similarité des traces, du contexte et du besoin d'analyse, entre l'analyse à réaliser dans l'expérimentation et celles qu'ils ont réutilisées. Pour ces trois éléments, nous avons utilisé une échelle de notation allant de 0 à 5, signifiant respectivement *Indépendant* et *Identique*. Concernant les traces, nous avons obtenu un degré de similarité estimé de 2.6 en moyenne $(1; 2; 3; 3; 4)$. Concernant le contexte, nous avons obtenu un plus fort besoin de similarité, avec une moyenne de 3.4 $(2; 3; 4; 4; 4)$. Finalement, concernant la similarité des besoins, nous avons obtenu une moyenne de 2.4 $(2; 2; 2; 3; 3)$.

Finalement, concernant l'implémentation d'une version adaptée du processus d'analyse narré, ou de certaines de ses étapes, nous avons constaté que les analystes ont eu plus de difficultés, comme l'atteste la Figure 14.4. Nous avons obtenu une moyenne de 1.8 sur une échelle de notation allant de 0 à 5, signifiant respectivement *Absolument pas implémenté* et *Totalement*.

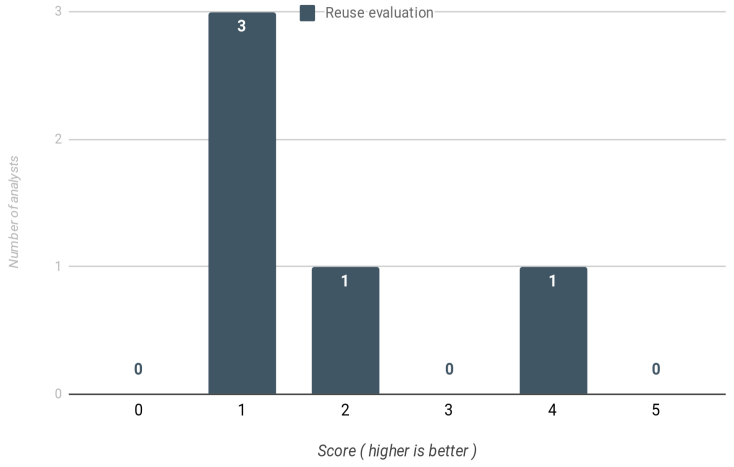


Figure 14.4.: Résultats des analystes concernant leur faculté à implémenter les analyses narrées qu'ils ont su adapter, d'après une échelle allant de 0 à 5, signifiant respectivement *Absolument pas* et *Totalement*.

14.2.5 Effet sur la capitalisation en général

Nous avons également collecté les retours des analystes concernant les aides spécifiques apportées par notre prototype **CAPTEN-TORTOISE**. Outre le fait que notre prototype a été un support lors de

l'élaboration de l'analyse (cf. Figure 14.3), nous pouvons extraire de ces retours trois principaux axes d'assistance.

Le premier concerne l'assistance dans la conception de l'analyse et la configuration de certaines de ses étapes. Par exemple, un analyste a répondu que cela lui a permis de "revoir [ses] paramétrages", deux analystes que cela leur a permis "d'élaborer de nouvelles classes" et un autre de "trouver des idées de prétraitements". Le deuxième axe concerne la compréhension de l'analyse et des besoins d'analyses auxquels elle répond. Par exemple, trois analystes ont répondu que le prototype leur a permis de "chercher de nouveaux [types de] résultats", et deux analystes ont répondu que **CAPTEN-TORTOISE** leur a permis "d'avoir une meilleure idée de ce qui était attendu [pour le besoin d'analyse]". Finalement, le troisième axe concerne l'assistance à propos de la qualité de l'analyse. Trois analystes s'en sont servi pour "avoir une méthode à laquelle [se] comparer".

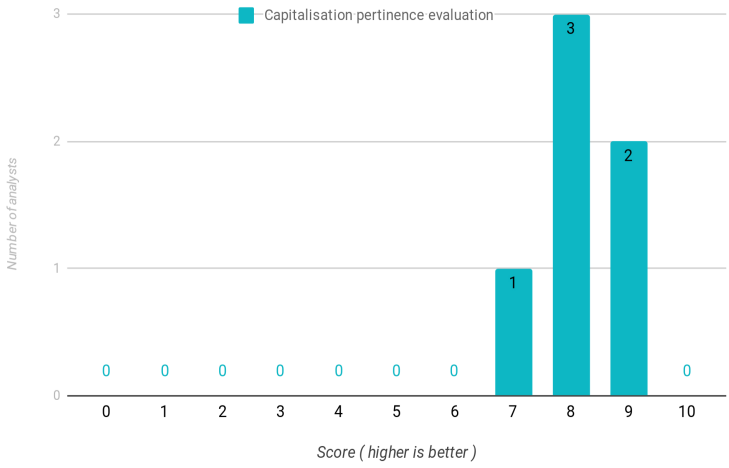


Figure 14.5.: Résultats de l'évaluation de la pertinence de notre approche comme une solution à la capitalisation des processus d'analyse de traces, allant de 0 à 10, signifiant respectivement *Inutile* et *Indispensable*.

Pour finir, les analystes ont donné leur avis global sur l'approche narrative proposée. On peut déjà remarquer que les différents éléments introduits par l'approche (*e.g.* processus d'analyse narrés, graphe de concepts, éléments narratifs) ont été compris par tous les analystes. Ils ont de plus estimé que cette approche permet de répondre à la problématique de la capitalisation au sein de la communauté des Learning Analytics. Les résultats obtenus avec une échelle de notation allant de 0 à 10, signifiant respectivement *Inutile* à *Indispensable*, sont visibles dans la Figure 14.5, avec une moyenne de 8. Tous ces résultats préliminaires semblent ainsi valider la pertinence de notre approche narrative pour la capitalisation des processus d'analyse de traces d'apprentissage.

Enfin, par rapport à une approche textuelle ou à une approche par workflows, les analystes ont jugé que notre approche leur apportait davantage. En effet, sur une échelle de 0 à 10 (signifiant respectivement *Aucun apport* et *Remplace totalement*), l'apport de notre prototype par rapport à une approche textuelle a été noté avec une moyenne de 8 $(7; 7; 8; 8; 8; 10)$, et avec une moyenne de 5.8 $(3; 5; 5; 7; 7; 8)$ par rapport à une approche par workflow. Les analystes ont jugé particulièrement intéressant de combiner dans notre approche à la fois les avantages de l'approche textuelle dans la précision que l'on peut apporter avec les éléments narratifs, et les avantages de l'approche workflows avec la structuration en étapes du processus. Ils regrettent cependant que, par rapport à l'approche workflows, nos processus d'analyse narrés ne soient pas exécutables.

14.3 Discussion

Ces résultats montrent que notre approche narrative pour la capitalisation des processus d'analyse de traces assiste la majorité des sujets pour adapter et réutiliser un processus d'analyse de traces existant.

Un point qui ressort de cette expérimentation est le fait que notre prototype – ou notre approche – n'ait rien apporté de plus à l'un des analystes lors de la troisième phase de l'expérimentation. Cette réponse est importante et nous a permis de préciser pour qui la capitalisation est importante dans le cycle d'élaboration d'une analyse.

En effet, le profil de ce sujet peut être assimilé à celui d'un analyste expert dans ce type d'analyse. Il avait déjà répondu à des besoins d'analyse très similaires (pour des jeux de données également similaires à celui donné lors de l'expérimentation), et avait donc déjà élaboré des analyses similaires. Il semble en effet probable que ces profils experts ne soient pas les premiers concernés par les propriétés de compréhension et de réutilisation offertes par la capitalisation. En revanche, ces analystes experts auront un rôle important dans la description narrée des processus d'analyse qu'ils mettent régulièrement en œuvre. Ces experts pourront également utiliser notre approche pour apporter des informations complémentaires ou des propositions d'amélioration sur des processus narrés décrits par d'autres analystes.

Un autre constat intéressant réside dans le fait qu'un des analystes a dit avoir choisi le besoin d'analyse de l'expérimentation car il avait une intuition concernant la technique d'analyse à mettre en œuvre pour y répondre. Ce comportement particulier nous laisse à penser que des analystes peuvent posséder des schémas d'analyse prédéfinis qu'ils sont capables de réutiliser, et qu'ils les spécialisent en fonction du contexte dans lequel ils se trouvent. Si c'est effectivement le cas, cela fait écho à la noèse que nous cherchons à capturer avec notre approche : bénéficier d'une approche narrative permettrait à de tels acteurs de pouvoir rechercher des analyses existantes à adapter, ce qui leur permettrait d'instancier ces schémas dans des contextes variés.

Nous avons également remarqué que, durant la phase qui consistait à mettre en œuvre le processus d'analyse sans utiliser notre prototype, aucun des six analystes n'a consulté l'existant, d'une manière ou d'une autre, alors que les besoins d'analyse proposés sont relativement fréquents. Ils ont seulement consulté le document explicatif des traces et deux d'entre eux ont également consulté la documentation de leur outil d'analyse. Or, cinq d'entre eux ont estimé, lors du questionnaire, que consulter l'existant en utilisant notre prototype les avait effectivement aidés à mettre en œuvre, ou à améliorer, leur processus d'analyse. De notre point de vue, ceci illustre bien le manque actuel d'outil de partage au sein de la communauté, voire un manque d'habitude au sein de la communauté à se référer à l'existant – symptomatique de l'état d'isolement des analyses. Cela appuie d'autant plus l'intérêt que pourraient avoir nos propositions dans la pratique des différents acteurs du cycle d'élaboration de l'analyse.

D'une manière générale, il est important de noter que **CAPTEN-TORTOISE**, étant un prototype de recherche, n'implémente pas toute notre ontologie, et que des défauts sur différents aspects, comme l'interface homme-machine, ont complexifié son utilisation et la compréhension de l'information présente. Par exemple, comme l'atteste le retour des utilisateurs, ces défauts expliquent majoritairement les résultats moins probants à propos de l'implémentation d'une version adaptée d'un processus narré (cf. Figure 14.4). Il s'avère que les difficultés rencontrées proviennent surtout d'un manque d'instructions dédiées à la manière d'implémenter ces processus d'analyses narrés dans des outils d'analyse conventionnels. En effet, même si notre ontologie prévoit d'associer à un opérateur narré des informations sur son implémentation dans divers outils d'analyse, ces informations ne sont pas encore gérées dans notre prototype.

Néanmoins, nous ne pouvons pas ignorer qu'il est possible que les utilisateurs aient subi une surcharge cognitive due à différents facteurs : la structuration en relations des différents éléments narratifs, le nombre de relations utilisées pour décrire l'information, et la difficulté d'appropriation par l'utilisateur des différents termes du vocabulaire contrôlé. Ceci aurait naturellement un effet sur les résultats concernant la compréhension, l'adaptation et la réutilisation des processus.

En effet, les utilisateurs avaient beaucoup de concepts à assimiler et à utiliser dans un temps assez court, l'expérimentation ayant une durée globale d'environ 3 heures. On peut par exemple remarquer qu'ils ont bien identifié l'utilité de la structuration en étapes d'un processus (cf. Tableau 14.1), mais que lors de cette première utilisation du prototype, ils n'ont peut-être pas eu l'occasion d'apprécier tout l'intérêt de la représentation des traces sous forme de graphe ou celui des éléments narratifs. Il est probable qu'une utilisation plus régulière de **CAPTEN-TORTOISE** permettrait une meilleure

compréhension des différents éléments de notre approche, et de leur intérêt pour l'adaptation et la réutilisation de processus d'analyse.

Enfin, bien que ces résultats préliminaires renforcent notre intuition concernant la forte dépendance des analyses par rapport au contexte dans lequel elles se trouvent (cf. Section 14.2.4), l'on peut constater que les analystes trouvent pertinent de réutiliser des processus d'analyse peu similaires pour ce qui est du besoin à satisfaire et des traces sur lesquelles porte l'analyse. Il nous semble que c'est l'approche narrative qui permet de compenser cette faible similarité, car ce sont majoritairement les éléments narratifs, en plus de la sémantique inhérente à notre approche, qui ont été utilisés par les analystes pour réaliser l'adaptation des processus à leur besoin.

Évaluation de la recherche sémantique

15

Sommaire

Section	**Introduction**	**189**
Section 15.1	**Scénarios d'usage**	**189**
Section 15.2	**Discussion**	**200**

Introduction

Ce chapitre concerne les évaluations que nous avons menées pour évaluer notre proposition théorique **CAPTEN-FRUIT** sur la recherche intelligente des processus d'analyse narrés et des étapes qui répondent à un besoin d'analyse. Il s'agit de tests menés suite à l'implémentation de notre approche dans notre prototype **CAPTEN-SEED**. Outre l'évaluation de cette proposition, l'objectif était de pouvoir illustrer les nouvelles possibilités offertes par notre approche ontologique des analyses.

Pour cela, ces tests ont été réalisés sous la forme de scénarios d'usage. Pour chacun, nous avons cherché à décrire un besoin d'analyse qui sera recherché dans l'ensemble des processus d'analyse narrés que nous avions préalablement décrits dans notre prototype (cf. Chapitre 11, page 153). Puis, nous avons étudié les résultats produits par notre mécanisme de recherche : nous voulions évaluer si les résultats proposés étaient identifiés correctement, et si leur pertinence vis-à-vis du besoin était correctement traduite [1].

De plus, nous procédons à une comparaison théorique des résultats que nous avons obtenus par rapport à ceux qui pourraient être obtenus avec d'autres outils d'analyse. À chaque fois, nous effectuons cette comparaison avec le ou les outils qui nous semblent être les plus pertinents.

Dans la suite de ce chapitre, nous présentons d'abord les scénarios d'usage que nous avons utilisés lors de nos tests, puis les résultats obtenus et les comparaisons. Nous concluons en discutant ces retours d'évaluation.

15.1 Scénarios d'usage

Dans cette section, nous présentons les différents scénarios d'usage utilisés pour tester notre proposition au travers de notre prototype, et présentons les principaux résultats obtenus. En plus de faire intervenir des besoins d'analyse différents, ces différents scénarios ont aussi pour but de concerner différents profils d'acteurs. Les scénarios d'usage ci-dessous sont classés par complexité croissante du besoin d'analyse décrit.

1. Un résultat A répondant mieux au besoin d'analyse qu'un résultat B doit avoir un score de pertinence $\mathfrak{p}_A$ supérieur à $\mathfrak{p}_B$, celui de B.

15.1.1 Récupérer les analyses concernant les apprenants : une dimension, un token

Description

Le premier scénario d'usage que nous avons mis en œuvre est le cas trivial où un acteur désire rechercher tous les processus narrés ou les étapes qui font intervenir la notion d'apprenant. Pour cela, la dimension de *desiderata* $\mathcal{D}_d$ du besoin d'analyse – qui permet de décrire son objectif – est exprimée uniquement avec le terme *Étudiant* de notre vocabulaire, tel que $\mathcal{D}_d = \langle \acute{E}tudiant \rangle$. L'objectif ici était d'étudier comment notre modèle se comporte lorsqu'un seul token [2] est recherché, notamment vis-à-vis de la couverture du besoin et de la pertinence des résultats.

De plus, utiliser le terme *Étudiant* n'est pas anodin : la majorité des analyses et des étapes concernent ce seul token ; il est facilement possible d'établir un inventaire pour vérifier la qualité de notre recherche. Enfin, cela nous permet aussi d'étudier si les termes en relation avec le terme d'*Étudiant* sont effectivement exploités. Cette notion de relation étant, rappelons-le, conditionnée par le patron de recherche associé à la dimension (ici de *desiderata*).

Bien qu'il s'agisse d'une recherche simple, il n'en reste pas moins qu'elle offre un cadre de sérendipité importante pour des décideurs ou des bénéficiaires de l'analyse. En effet, cela leur permet de récupérer tout l'existant et d'en prendre connaissance pour ainsi apporter des pistes de ce qu'il est possible de réaliser lors d'analyses dans notre discipline – et donc servir de base lors de la réflexion.

Résultat

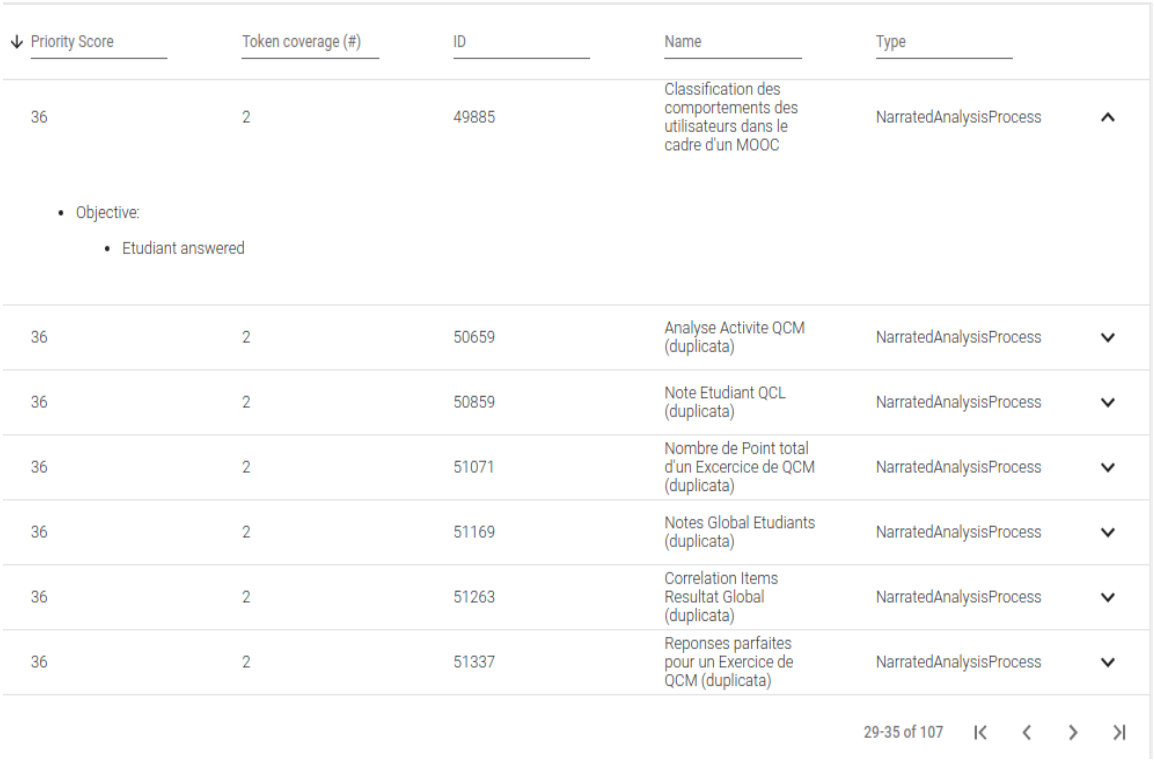
Après avoir requêté nos processus lors de ce scénario d'usage, nous avons récupéré tous les processus d'analyse narrés qui étaient en relation directe avec ce concept d'*Étudiant* – ou subsumés avec des termes comme *Apprenant*. Un aperçu des résultats est visible dans la Figure 15.1. Nous avons également été en mesure de récupérer toutes les étapes des processus d'analyse narrés qui faisaient intervenir ce concept.

Nous avons aussi jugé que la pertinence calculée pour les résultats était satisfaisante et représentative. En effet, même si une étape était identifiée comme un candidat potentiel par le système, ce dernier a considéré les processus d'analyse narrés comme plus à même de répondre au besoin d'analyse énoncé (*i.e.* ils possédaient un score de pertinence plus élevé que les étapes). Cela s'explique par la manière dont le score de pertinence est construit : il repose sur la structure même des opérations narrées et sur le patron de recherche. Un processus narré étant composé d'étapes, et d'éléments narratifs, le score potentiel sera naturellement plus important. De plus, le score est calculé en fonction du nombre de types d'éléments du patron de recherche qui possèdent le token recherché.

Nous n'avons pas identifié de faux positifs. Concernant les informations associées au backtrack, et cela concerne également les autres scénarios d'usage, elles étaient correctes mais présentées avec beaucoup de redondance. La faute principalement aux graphes de variables d'entrée et de sortie qui sont, dans **CAPTEN-TORTOISE**, considérés comme différents pour des raisons d'implémentation. En effet, lorsqu'un graphe de variables de sortie est utilisé en entrée, nous le copions. L'objectif ici était de pouvoir rendre à la fois les étapes liées entre elles, mais aussi indépendantes, ceci afin de pouvoir les réutiliser seules : nous planifions déjà de pouvoir assister les utilisateurs en leur fournissant des blocs d'étapes – n'appartenant pas forcément au même processus d'analyse – qui répondent au besoin d'analyse exprimé.

Théoriquement, cette recherche peut s'appliquer à la majorité des outils d'analyse existants qui proposent une option de recherche. Néanmoins, la qualité des résultats va dépendre directement de l'endroit où le token – dans le cas de ces outils, il s'agit alors d'une simple chaîne de caractères – est recherché (*e.g.* le nom du processus). De plus, il est important de noter que tous les éléments

2. Les tokens sont des concepts de l'ontologie – des classes ou des propriétés sémantiques – qui représentent un point spécifique d'une dimension spécifique (cf. Équation 3.2, page 135).



Priority Score	Token coverage (#)	ID	Name	Type
36	2	49885	Classification des comportements des utilisateurs dans le cadre d'un MOOC	NarratedAnalysisProcess
• Objective: • Etudiant answered				
36	2	50659	Analyse Activite QCM (duplicata)	NarratedAnalysisProcess
36	2	50859	Note Etudiant QCL (duplicata)	NarratedAnalysisProcess
36	2	51071	Nombre de Point total d'un Excercice de QCM (duplicata)	NarratedAnalysisProcess
36	2	51169	Notes Global Etudiants (duplicata)	NarratedAnalysisProcess
36	2	51263	Correlation Items Resultat Global (duplicata)	NarratedAnalysisProcess
36	2	51337	Reponses parfaites pour un Exercice de QCM (duplicata)	NarratedAnalysisProcess

29-35 of 107

Figure 15.1.: Capture d'écran de notre prototype de recherche montrant un sous-ensemble des résultats obtenus.

faisant intervenir un Apprenant et non pas directement un Étudiant ne seront pas considérés comme de potentiels résultats dans des systèmes sans dictionnaire ou taxinomie. Ainsi, nous pensons que des outils comme UnderTracks ou Galaxy pourraient fournir un sous-ensemble de nos résultats. En revanche, kTBS pourrait tenir compte de la particularité entre Apprenant et Étudiant puisqu'il utilise également une approche de type web sémantique, bien qu'il n'existe aucun mécanisme de recherche en soi – elle s'opère via des requêtes SPARQL. Aussi, au même titre que R, il n'est pas pertinent de le comparer à des outils proposant une abstraction des mécanismes de recherche, comme UnderTracks ou notre approche, forcément plus limité, mais cherchant à inclure et concerner plus d'utilisateurs.

15.1.2 Calculer la proportion de succès d'un QCM : une dimension, deux tokens

Description

Le deuxième scénario d'usage que nous avons mis en œuvre est le cas où un acteur recherche comment obtenir la proportion de succès dans un QCM. Il s'agit d'un scénario où l'on introduit deux tokens pour décrire le besoin d'analyse. Ainsi, nous avons exprimé la dimension de *desiderata* comme $\mathcal{D}_d = \langle Proportion\_de\_Succes; QCM \rangle$, avec les termes toujours issus de notre vocabulaire. L'objectif principal ici était de tester la recherche multi-tokens sur une seule dimension, d'observer les résultats produits et d'évaluer si la couverture du patron de recherche pour la dimension *desiderata* était suffisante.

Nous pensons que ce type de scénario convient bien à des profils d'acteur comme des bénéficiaires ou des décideurs. En effet, il permet d'exprimer un besoin d'analyse simple et de consulter l'existant en fonction.

Résultat

Lors de la recherche, le système a été en mesure de récupérer tous les processus d'analyse narrés et les étapes qui couvraient le besoin d'analyse, validant ainsi notre algorithme dans le cas d'une approche par plusieurs tokens. Les résultats que nous avons obtenus étaient l'union des résultats pour le concept de *Proportion_de_Succes* et de *QCM*. Aussi, les processus d'analyse narrés qui n'étaient en lien qu'avec des QCM ou des propotions de succès ont moins bien été classés en matière de pertinence que ceux couvrant les deux concepts. Le même constat s'applique pour les étapes. La Figure 15.2 propose un aperçu des résultats obtenus, notamment un processus d'analyse qui répond aux deux tokens recherchés ; l'on y observe notamment que son score de pertinence est supérieur aux deux autres analyses présentes.

De plus, il est à noter que certaines étapes des processus d'analyse narrés résultats étaient mieux notées que certains processus d'analyse narrés eux-mêmes. En effet, elles répondaient aux deux tokens posés en question, comme la dernière étape du processus calculant la proportion de succès pour un QCM.

Nous n'avons pas non plus constaté de faux positifs lors de cette recherche. Des étapes d'analyse étaient en revanche bien moins pertinentes que ne pouvait le laisser paraître leur score de pertinence, principalement car elles n'étaient simplement qu'en relation avec des graphes de variables portant ces concepts. Par exemple, une étape calculant le nombre total d'apprenants – appartenant au processus résultat – a été estimée pertinente car le terme QCM était présent dans son graphe de variables. La prise en compte de l'importance des éléments du patron de recherche lors du calcul du score devrait pouvoir solutionner ce problème en pondérant cette pertinence (cf. Définition 3.3, page 136).

Théoriquement, il est encore possible d'obtenir des résultats avec les outils d'analyse classiques qui soient plus ou moins conformes à nos résultats, avec les mêmes lacunes que celles citées précédemment. En sus de cela, il faut également prendre en compte comment l'outil effectue la recherche lorsque la chaîne de caractères est composée de plusieurs termes. Soit il effectue une recherche exacte dans la chaîne[3], soit il recherche l'occurrence d'un ou plusieurs termes. À ce niveau, nous pensons que RapidMiner, Orange : Data Mining, ou encore UnderTracks fourniraient des résultats, bien qu'incomplets. Les meilleurs résultats seront fournis par Galaxy ou Taverna, qui permettent de décomposer le processus un peu plus finement – et donc d'opérer une recherche sur certaines étapes. Toutefois, il n'existera pas, à notre connaissance, de classement effectif des résultats en fonction de leur pertinence.

15.1.3 Catégoriser les apprenants : une dimension, une relation

Description

Dans ce scénario d'usage, il s'agit de consulter l'existant pour trouver des processus d'analyse qui catégorisent des étudiants. Il s'agit ici d'un scénario où l'on va tester la description d'une seule dimension *via* une relation pour trouver des éléments solutions. À noter que ce scénario d'usage est une requête relativement courante. Elle peut être spécialisée par l'utilisation de la dimension contextuelle, ceci afin de préciser certaines contraintes importantes. C'est d'ailleurs le cas pour le cinquième scénario d'usage.

Aussi, nous avons défini la dimension de *desiderata* comme $\mathcal{D}_d = \langle categorise(Analysis, Apprenant) \rangle$. L'objectif ici était de tester la recherche *via* une relation et d'évaluer si la couverture du patron de recherche était toujours suffisante. Nous testions également l'effet des propriétés sémantiques lors de la recherche.

Nous pensons que ce scénario d'usage convient mieux à un profil de chercheurs et d'analystes, notamment du fait de la question ouverte posée ici. En effet, une telle requête permet de récupérer l'état de l'art en ce qui concerne la catégorisation d'apprenants.

3. Le système ne considère alors comme résultat que des chaînes de caractères contenant *Proportion_de_succes QCM*.

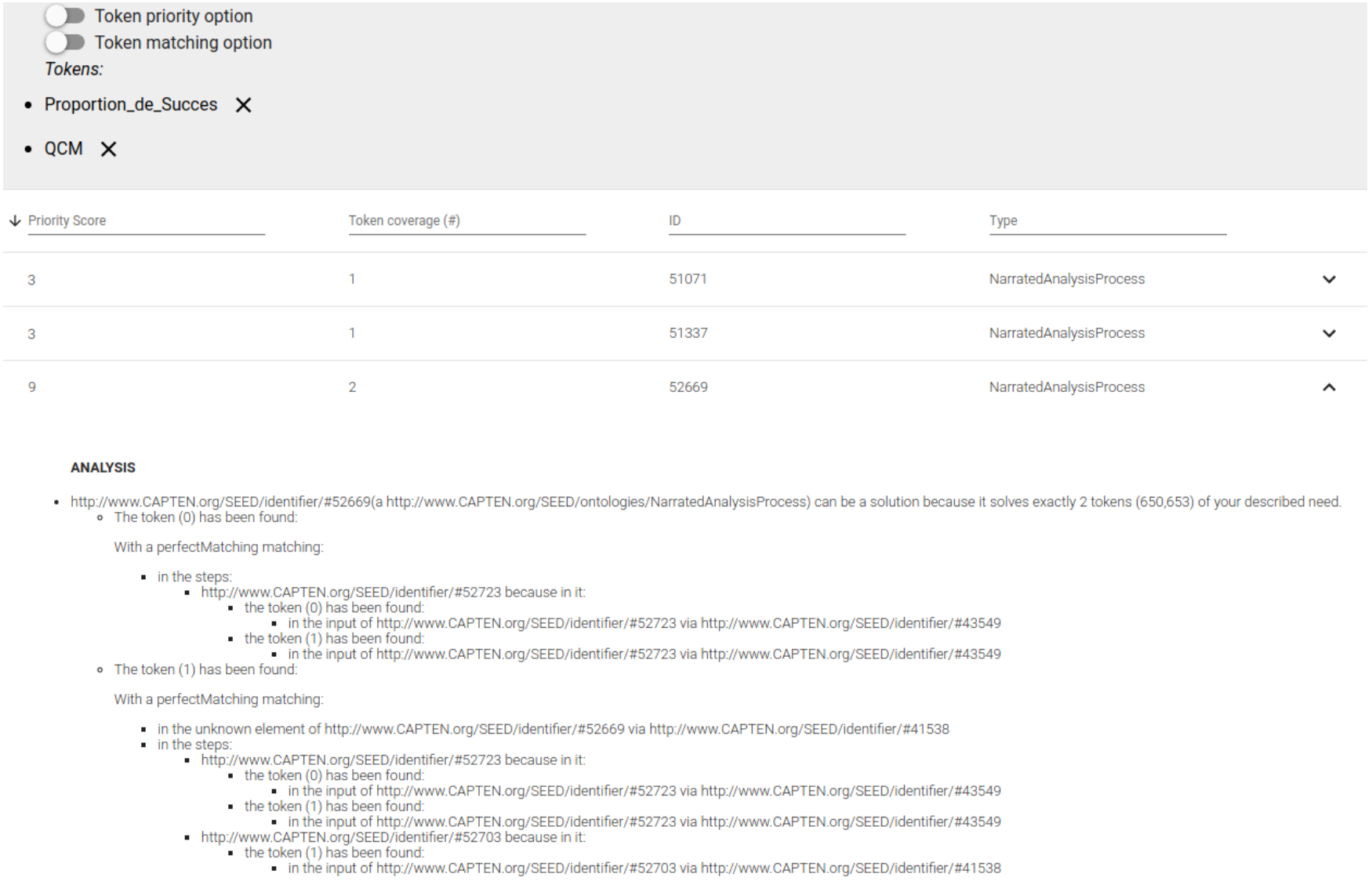


Figure 15.2.: Capture d'écran de notre prototype de recherche montrant un sous-ensemble des résultats obtenus pour la dimension de *desiderata*. Un processus répondant aux deux tokens est affiché en détail pour montrer les informations expliquant le score du processus, obtenus *via* backtrack.

Résultat

Lors de cette recherche, le système a correctement identifié dans notre prototype les deux seuls processus d'analyse narrés – sur les neuf décrits (cf. Section 11.3, page 159) – qui effectuaient une catégorisation des apprenants. Une des deux analyses a été jugée plus pertinente que l'autre, notamment car le nombre de types d'éléments du patron de recherche ayant été identifiés a été plus important. Cela est dû à une étape portant aussi cette relation. D'ailleurs, le système a également identifié cette étape comme candidat potentiel, avec un score moindre. Les résultats sont visibles dans la Figure 15.3.

De plus, l'identification de ces processus d'analyse narrés ne pouvait se faire qu'en exploitant des propriétés sémantiques de notre ontologie comme la transitivité des relations. Par exemple, l'un des processus d'analyse identifié comme candidat concerne les étudiants et non les apprenants. Dans notre ontologie, la classe *Etudiant* est subsumée par la classe *Apprenant*.

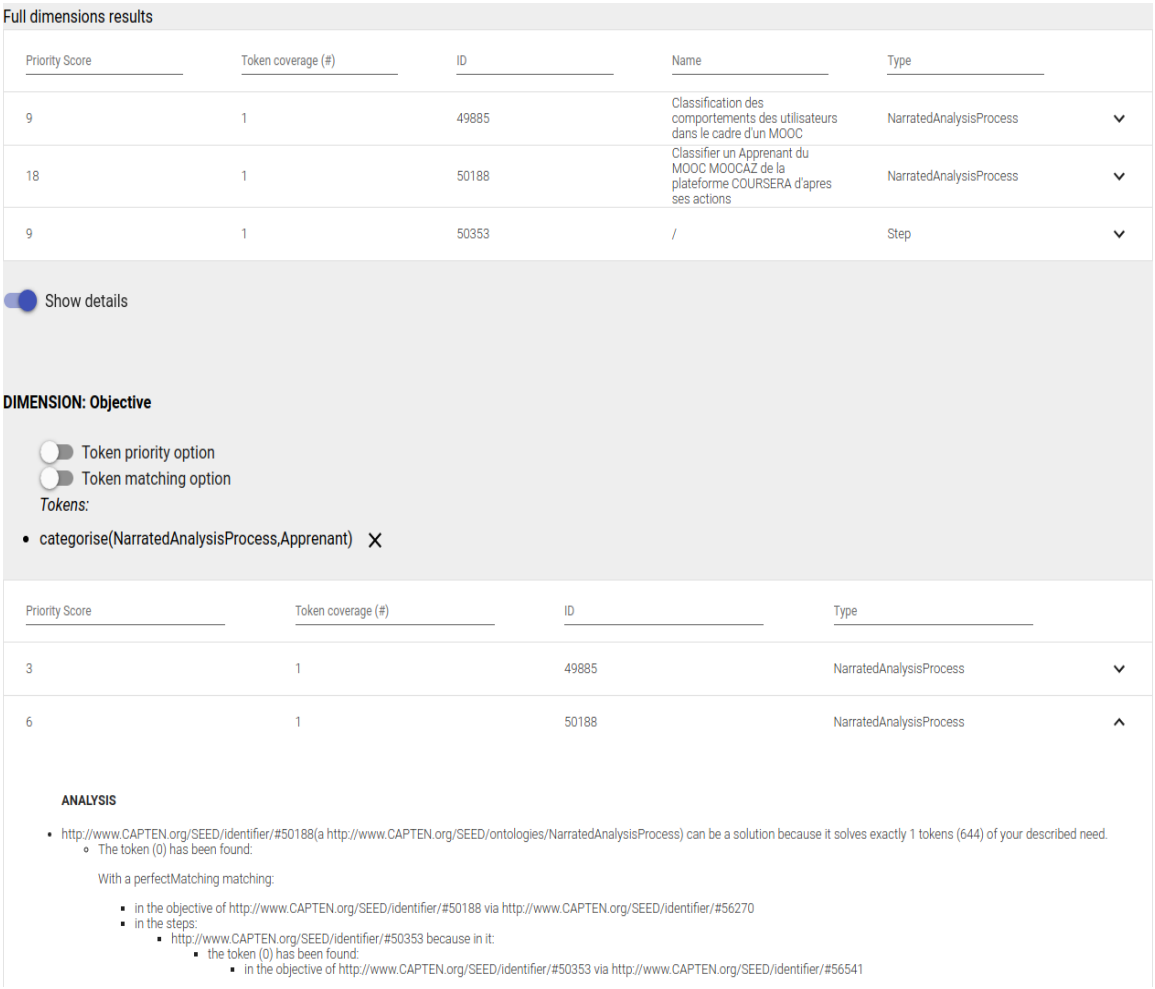


Figure 15.3.: Capture d'écran de notre prototype de recherche montrant les résultats obtenus pour la dimension de *desiderata*, en recherchant une relation. Le processus le plus pertinent est affiché en détail pour observer l'explication générée par le système, suite au backtracking.

Ces résultats nous laissent penser que notre patron de recherche offre une couverture suffisante pour identifier les processus d'analyse à partir de leur objectif. De surcroît, le mécanisme de recherche basé sur les relations apparaît cohérent et bien supporté par notre proposition. La relation permet de spécifier des contraintes supplémentaires entre les éléments lors de la recherche, qu'il n'est pas possible de représenter avec des tokens simples. De plus, nous n'avons pas eu de faux positifs.

À notre connaissance, aucun outil d'analyse actuel n'est en mesure d'effectuer ce type de requêtes relationnelles, même ceux faisant intervenir pour certains aspects des ontologies (*e.g.* Research Object). *A minima*, et nous le verrons dans le scénario d'usage suivant, l'on pourrait découper cette relation en chaînes de caractères pour effectuer cette recherche dans ces outils, au détriment de la pertinence des résultats.

15.1.4 Catégoriser les apprenants : une dimension, une relation décomposée

Description

Ici, nous reprenons le scénario d'usage précédent, à savoir trouver des résultats qui catégorisent des apprenants. La différence est que nous n'avons pas utilisé de relation pour exprimer la dimension de *desiderata*, uniquement des tokens, ce qui s'assimile à utiliser la relaxe $\mathfrak{Q}_{10}$ (cf. Section 9.3, page 134). Autrement dit, $\mathcal{D}_d = \langle Catégoriser; Analysis; Apprenant \rangle$. L'objectif était d'observer si, entre une recherche par relation et un ensemble des trois tokens simples, une variation au niveau des résultats pouvait être constatée. De plus, cela nous permettait d'évaluer plus en détail l'évolution des scores de pertinences.

Bien que ce scénario soit similaire au scénario d'usage précédent, nous pensons ici qu'un profil moins expert convient mieux, comme un bénéficiaire. En effet, être en mesure d'exprimer correctement un besoin d'analyse peut s'avérer être une étape complexe.

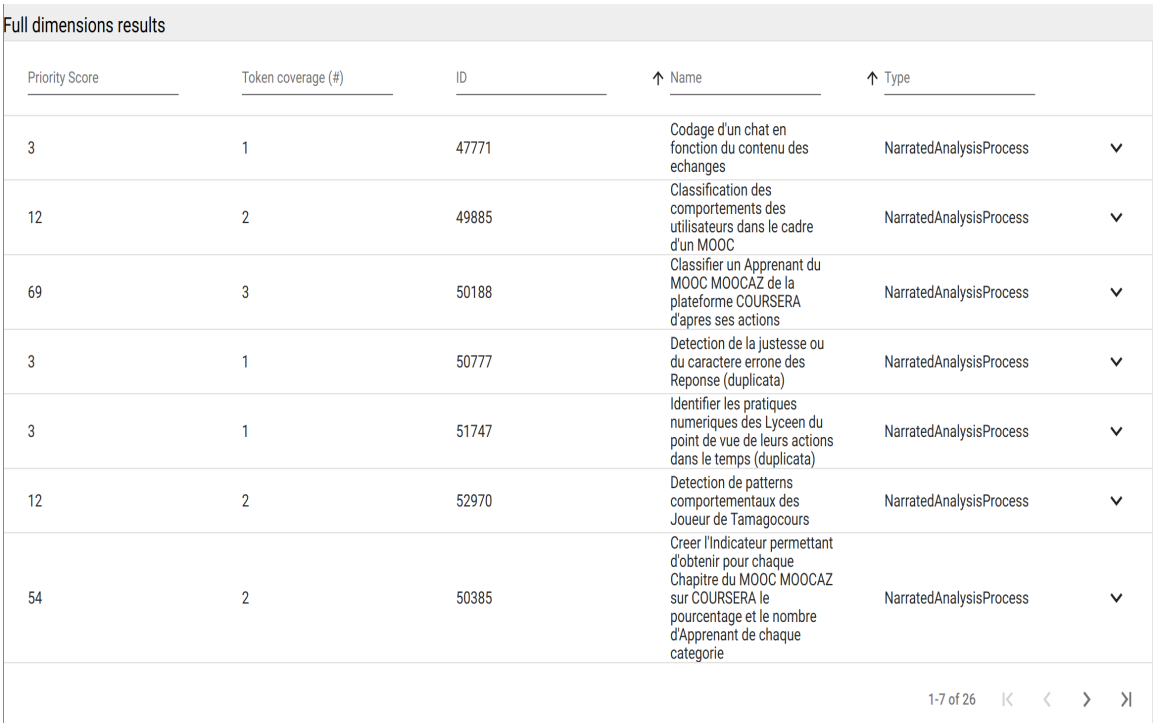
Résultat

Lors de cette recherche, le système a été en mesure de récupérer tous les processus d'analyse narrés et les étapes qui couvraient le besoin d'analyse énoncé avec les trois concepts ci-dessus. La Figure 15.5 propose un aperçu des résultats. Comparativement au scénario d'usage précédent, nous avons ici été en présence de faux positifs à cause de l'utilisation du concept *Analysis*. Ce dernier faisant référence au processus d'analyse narré, le système a ainsi identifié tous nos processus d'analyse narrés comme de potentiels candidats : ils répondent en effet tous à l'un des trois tokens recherchés. De plus, nous avons aussi constaté que ces processus étaient parfois classés comme des candidats équivalents, voire meilleurs, que des étapes, alors qu'aucune catégorisation n'était effectuée.

Pour pallier cette mauvaise classification du système, et essayer de nous rapprocher des résultats obtenus dans le scénario précédent, nous avons utilisé les modificateurs de scores (cf. Section 12.1, page 164). Néanmoins, après avoir diminué l'importance du concept *Analysis* et avoir fait émerger les deux analyses précédentes comme les meilleurs résultats, certains faux positifs avaient encore leur score de pertinence élevé par rapport à d'autres étapes plus pertinentes à notre sens. Cela s'explique par le fait que le concept d'*Analysis* pesait tout de même dans le score : à couverture égale, une analyse sera toujours considérée par le système avec une pertinence plus élevée. Pour remédier à ce problème, il faut pouvoir donner à l'utilisateur la possibilité de ne plus faire peser le token dans la recherche – nous avons implémenté cette fonctionnalité simplement en permettant de le supprimer.

Ces résultats montrent l'importance de la description du besoin d'analyse et de la nécessité pour l'utilisateur d'avoir conscience de la couverture sémantique qu'il recherche lorsqu'il utilise un token. En effet, les variations semblent importantes entre l'utilisation d'une relation et d'un ensemble de trois termes, principalement car la couverture du système pour le dernier cas s'apparente à une union de la couverture individuelle de chacun des termes. Dès lors, une solution pourrait être d'utiliser un langage naturel contrôlé pour décrire l'effet du token dans la recherche, à l'instar des travaux comme SQUALL (Ferré, 2012) ou SPARE-LNC (Kong Win Chang et al., 2015) qui permettent d'écrire des requêtes en langage naturel.

À titre de comparaison avec les autres outils, nous pouvons ici nous replacer dans le cas du deuxième scénario d'usage, étendu ici avec trois tokens au lieu de deux. Ces tokens pourraient être décrits



Full dimensions results

Priority Score	Token coverage (#)	ID	Name	Type
3	1	47771	Codage d'un chat en fonction du contenu des echanges	NarratedAnalysisProcess
12	2	49885	Classification des comportements des utilisateurs dans le cadre d'un MOOC	NarratedAnalysisProcess
69	3	50188	Classifier un Apprenant du MOOC MOOCAZ de la plateforme COURSERA d'apres ses actions	NarratedAnalysisProcess
3	1	50777	Detection de la justesse ou du caractere errone des Reponse (duplicata)	NarratedAnalysisProcess
3	1	51747	Identifier les pratiques numeriques des Lyceen du point de vue de leurs actions dans le temps (duplicata)	NarratedAnalysisProcess
12	2	52970	Detection de patterns comportementaux des Joueur de Tamagocours	NarratedAnalysisProcess
54	2	50385	Creer l'Indicateur permettant d'obtenir pour chaque Chapitre du MOOC MOOCAZ sur COURSERA le pourcentage et le nombre d'Apprenant de chaque categorie	NarratedAnalysisProcess

1-7 of 26

Figure 15.4.: Capture d'écran de notre prototype de recherche montrant les premiers résultats obtenus suite à la recherche du besoin d'analyse. Le score de pertinence permet ici d'identifier les processus répondant le mieux au besoin d'analyse.

par une chaîne de caractères de trois termes distincts qui seraient alors consommés par le moteur de recherche, le comportement de recherche dépendant encore de l'outil (*i.e.* exacte ou d'après les occurrences des termes). Nous retrouvons aussi les mêmes lacunes que celles précédemment citées (*e.g.* résultats incomplets), en plus de l'impossibilité de fournir une importance aux différents termes utilisés. Nous pensons néanmoins que Taverna produirait les meilleurs résultats parmi ces outils, grâce à la fois aux métadonnées associées aux processus, à la possibilité d'utiliser des filtres lors de la recherche, et au fait que le nom des opérateurs et certaines métadonnées (*e.g.* tags, citations, review) sont également pris en compte lors de la recherche.

15.1.5 Classifier les apprenants dans un jeu sérieux : deux dimensions, une relation, un token

Description

Dans ce scénario d'usage, il s'agit de rechercher les processus d'analyse narrés qui proposent une classification des apprenants, le tout contextualisé dans le cadre d'un jeu sérieux. Ce scénario a été utilisé pour décrire notre prototype (cf. section 12.1, page 164). L'objectif était d'introduire une nouvelle dimension lors de la recherche, d'une part pour vérifier la couverture du patron de recherche du contexte et les résultats produits, et d'autre part pour étudier si l'agrégation des résultats de chaque dimension était cohérente.

Comme vu précédemment, le besoin d'analyse de ce scénario d'usage s'exprime sous la forme d'une dimension de *desiderata* $\mathcal{D}_d = \langle classify(Analysis, Apprenant) \rangle$, et d'une dimension contextuelle $\mathcal{D}_c = \langle Serious\_Game \rangle$.

Ce scénario d'usage concerne principalement les analystes.

Résultat

Lors de la recherche, le système a correctement identifié le processus d'analyse narré qui devait être identifié comme candidat potentiel (cf. Figure 12.1, page 165). Pour trouver ce processus d'analyse, le système devait là encore exploiter des relations sémantiques, par exemple transitives, qui s'appliquaient surtout à la propriété *classify*. Toutefois, nous n'avons pas pu travailler avec des relations d'équivalence entre les propriétés de notre ontologie pour des raisons techniques. Nous avons simulé ce comportement en utilisant une approche taxinomique, mais cela réduit tout de même les possibilités d'inférences du système – et pourrait causer des aberrations sémantiques, comme l'apparition de cycles de subsomption [4], et associer de mauvaises propriétés aux classes de l'ontologie. Cette limitation technique est due au raisonneur que nous utilisons, qui était dans l'impossibilité de définir ce type de relation (TERDJIMI, 2019). En effet, il s'agit d'un raisonneur OWL Light qui ne permet pas d'avoir une expressivité OWL Full, notamment pour se prémunir de la complexité que cela engendre et des risques d'indécidabilité.

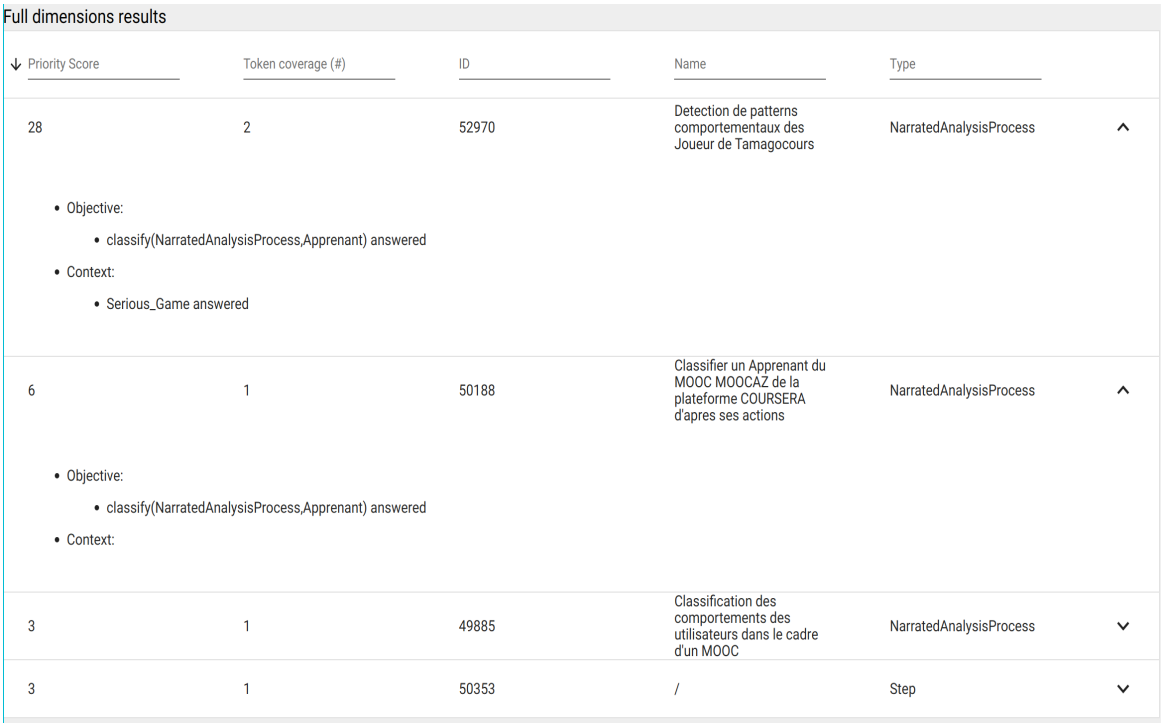


Figure 15.5.: Capture d'écran de notre prototype de recherche montrant les premiers résultats obtenus suite à la recherche du besoin d'analyse. Le premier processus d'analyse narré couvre ici l'entièreté du besoin d'analyse, le deuxième ne couvre que la dimension de *desiderata*. Le système considère ainsi que le premier processus est le plus pertinent.

Nous notons cependant la présence de faux négatifs dans notre recherche. Nous attendions les étapes du processus d'analyse candidat également comme des candidats potentiels, mais cela n'a pas été le cas. En effet, ces étapes auraient dû émerger comme potentiels candidats dans la dimension contextuelle lors de la recherche. La raison supposée de ces faux négatifs est que, outre des problèmes de description dans les processus d'analyse narrés (cf. la discussion suivante), certaines règles concernant les requêtes implémentées doivent être corrigées pour appliquer transitivement le contexte d'une analyse, porté par un élément narratif de type contexte, à toutes ses étapes.

Théoriquement, et à notre connaissance, ce type de recherche ne peut pas être correctement conduit dans les outils d'analyse. En effet, il fait à la fois intervenir des notions de relations fortes entre les différents éléments à rechercher, et une notion de contexte qui doit être prise en compte par le système

4. Un exemple : Soit A, B, C des classes, b une propriété partant de B. Si l'on a $C \sqsubset B \sqsubset A$ et que l'on indique une "pseudo-équivalence" entre A et C, tel que $A \sqsubset C$, un non-sens apparaît : la classe A sera probablement attachée à la relation b, et la classe B n'a d'autre choix que d'être "égale" à A.

lors de la recherche. Un outil d'analyse doté de métadonnées comme Taverna chercherait – encore une fois sous forme textuelle – cette notion de contexte partout dans le processus d'analyse avec une chaîne de caractères, au même titre que les tokens de la dimension de *desiderata*, comme nous l'avons vu précédemment. Un tel outil prend alors le risque de faire apparaître de faux positifs à l'utilisateur – sans pouvoir lui donner d'explication.

15.1.6 Identifier des régularités dans les actions des apprenants : deux dimensions, quatre relations

Description

Enfin, dans ce dernier scénario d'usage, il s'agit de découvrir dans l'existant des régularités dans les actions des apprenants, dans le contexte d'un MOOC. Ce besoin d'analyse plus complexe se matérialise sous la forme d'une dimension $\mathcal{D}_d = \langle decouvre(Analysis, Parcous), faitPar(Parcous, Apprenant), decouvre(Analysis, Pattern)\rangle$ et une dimension contextuelle $\mathcal{D}_c = \langle concerne(Analysis, MOOC)\rangle$.

L'objectif ici était de tester si la recherche s'opérait correctement avec des dimensions complexes, et si les scores de pertinence des résultats représentaient bien leur pertinence effective.

Résultat

Lors de cette recherche beaucoup plus complexe, le système a réussi à identifier le processus d'analyse narré que nous attendions comme seul résultat. Pour identifier ce processus d'analyse, le système devait utiliser la similarité existant entre deux classes de l'ontologie : *Parcours* et *Pattern*. Notre prototype ne permettant pas encore vraiment de déclarer la similarité entre deux termes du vocabulaire, nous avons donc utilisé la relation *assimilableA* pour traduire cette similarité relative. Les tokens utilisés, ainsi que le résultat de la recherche, sont visibles dans la Figure 15.6.

Ainsi, cela laisse penser que notre approche s'adapte à des besoins d'analyse complexes, dans la mesure où ils sont correctement décrits. Notre approche évite, par l'entremise des relations, d'obtenir des candidats qui ne correspondent pas au besoin d'analyse décrit. Toutefois, il est possible d'étendre ces résultats en assouplissant les règles de recherche, par l'application de moins de contraintes *via* les relaxes de classes.

Toutefois, la présence de faux négatifs pour la dimension contextuelle est à noter. Nous n'avons pas réussi à récupérer en candidat potentiel les analyses destinées à un contexte de MOOC – ce qui aurait dû être le cas. Au vu des bons résultats produits par la dimension de *desiderata*, nous pensons que ces faux négatifs peuvent être induits par deux facteurs. Le premier est, comme cité précédemment, lié aux règles de requêtes implémentées qui ne tiennent pas correctement compte des relations ontologiques ; le deuxième facteur peut être lié à une description incomplète des analyses dans notre cadre narratif – par exemple, certains éléments narratifs manquent, ou certaines relations entre les éléments de l'analyse manquent.

Enfin, nous ne pensons pas qu'il soit possible d'exprimer une telle requête dans les outils d'analyse traditionnels tout en s'assurant d'obtenir un sous-ensemble de résultats cohérents. Cela fait directement suite à tous les constats précédemment évoqués dans les différents scénarios d'usage.

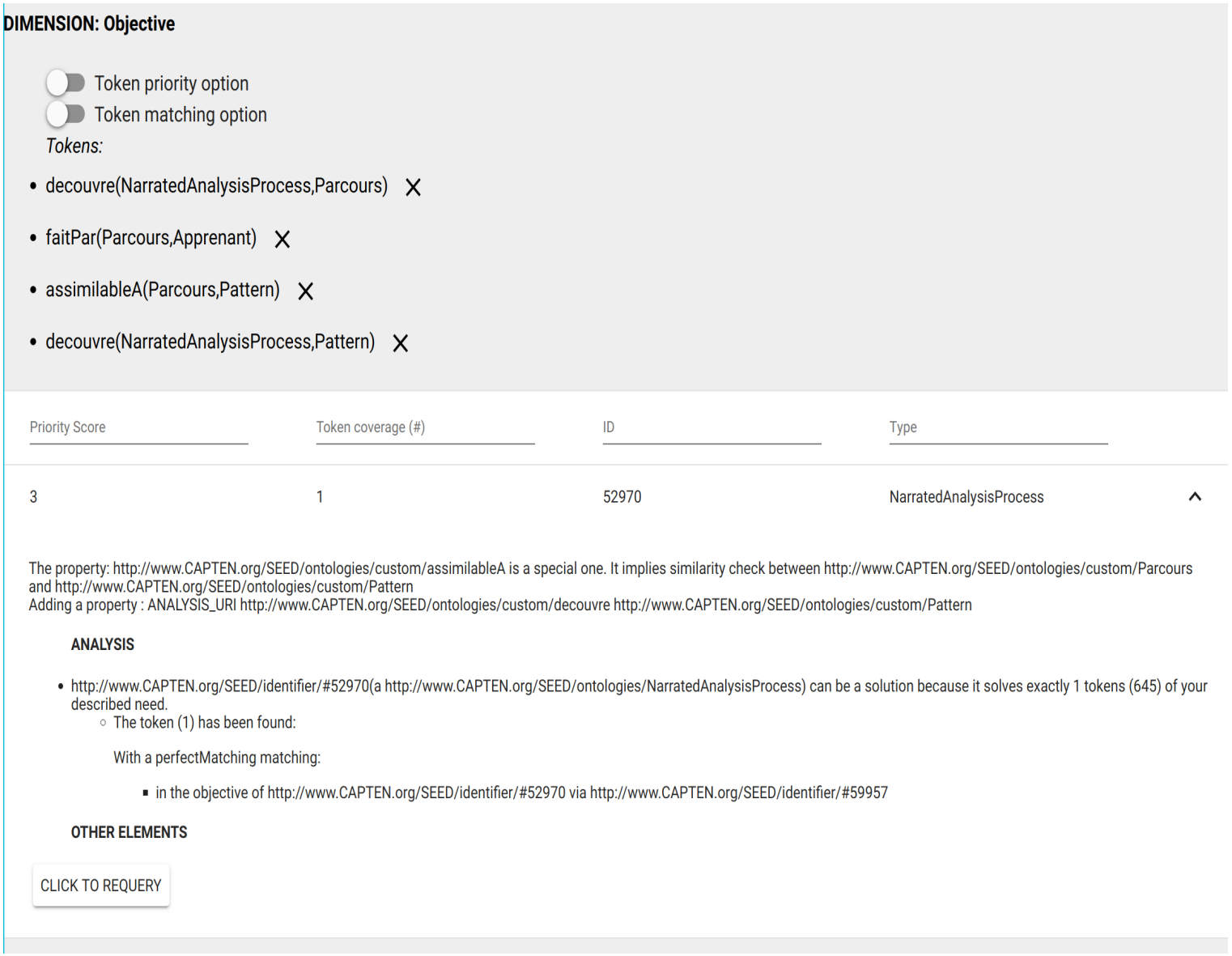


Figure 15.6.: Capture d'écran de notre prototype de recherche montrant les tokens *relations* utlisés lors de la recherche et le résultat obtenu pour la dimension de *desiderata*.

15.2 Discussion

Bien qu'il s'agisse ici de tests, nous avons jugé important de les inclure dans ce manuscrit. Cela permet d'une part d'illustrer l'utilisation de notre ontologie pour la narration des processus d'analyse, et d'autre part de souligner l'importance de leur bonne description dans notre framework et de justifier l'effort associé à cette description. Cela nous permet aussi d'introduire des perspectives en lien.

Globalement, nous avons eu de bons résultats préliminaires concernant cette recherche intelligente. Nous avons dans l'ensemble réussi à obtenir des candidats pertinents et correspondant à nos attentes, et limité la présence de faux positifs. Nous avons également pu apporter des éléments de validation du patron de recherche associé à la dimension de desiderata, du mécanisme de relaxe de contraintes et du calcul du score, dans une certaine mesure.

Toutefois, nous pensons important de souligner que ces tests ne sont que les prémisses à une vraie validation de notre proposition de recherche intelligente. En effet, ces tests sont biaisés par le fait que les scénarios d'usage sont appliqués sur une population de processus d'analyse faible, que nous avons nous-même définis : les besoins d'analyses ont ainsi potentiellement été influencés.

Ils sont aussi biaisés par le fait qu'ils sont réalisés dans notre prototype **CAPTEN-TORTOISE**, qui n'implémente pas entièrement notre ontologie, nous obligeant certaines fois à modifier les requêtes à appliquer. De surcroît, nous avons constaté à certains moments des erreurs dans la description ontologique de ces processus d'analyse narrés (*e.g.* relation sémantique non correctement référencée), introduites notamment lors de leur export par le prototype. La conséquence directe de tout cela est d'affecter le raisonnement conduit par le prototype et nous pensons qu'il s'agit d'une cause plausible pouvant expliquer la présence des faux négatifs.

Il convient donc de réaliser des expérimentations qui s'ancrent dans un contexte où de nombreux processus d'analyse sont narrés par différents acteurs. En plus de permettre d'évaluer de manière approfondie les différents éléments de nos approches, cela permettrait aussi d'étudier l'effet de la manière de narrer un processus sur leur recherche.

Enfin, dans ces tests, nous n'avons introduit que deux dimensions : celle de *desiderata* et de *contexte*. Il serait aussi intéressant d'étudier comment notre proposition de recherche intelligente se comporte avec un plus grand nombre de dimensions, notamment concernant la classification des candidats potentiels et la possibilité pour l'utilisateur d'explorer l'ensemble de ces candidats pour identifier manuellement ceux qui lui conviennent.

De plus, introduire de nouvelles dimensions nous permettrait d'exploiter plus encore les éléments narratifs, et l'effet des opérateurs dans les étapes. Actuellement, nous avons limité ces points dans nos tests, principalement pour pouvoir étudier l'effet des différents aspects de notre proposition (*e.g.* similarité, modificateur). Cependant, davantage de dimensions représenteraient des sources d'information majeures pour affiner la pertinence des résultats vis-à-vis du besoin d'analyse.

Partie V

In Fine

Conclusion

> *Le clou du pestacle !*
>
> —

Dans le cadre de cette thèse en informatique à la croisée des domaines des Environnements Informatiques pour l'Apprentissage Humain (EIAH), des Learning Analytics et de l'Ingénierie des Connaissances, nous avons abordé la question de la capitalisation des processus d'analyse de traces d'apprentissage. L'objectif de la capitalisation des processus d'analyse est de permettre aux différents acteurs de l'analyse de disposer des analyses existantes – déjà réalisées par les analystes et pour d'autres situations pédagogiques, pour les consulter, les comprendre, et éventuellement les adapter pour les réutiliser afin de répondre à leur propre besoin d'analyse.

Notre recherche consistait à mener un travail de fond sur cette question de la capitalisation des processus d'analyse de traces pour proposer des modèles capables de la rendre effective. Nous souhaitions pouvoir décrire les processus d'analyse de telle sorte qu'ils soient facilement adaptables à d'autres contextes pédagogiques, et les rendre accessibles et utilisables par toute la communauté, pour permettre à tous ses acteurs de s'impliquer pleinement et pour pouvoir les assister. Ainsi, et en tant que proposition préliminaire à ces modèles, nous avons proposé un cycle d'élaboration et d'exploitation de l'analyse, et une définition de la capitalisation. Nous la définissons comme un ensemble de six propriétés hiérarchiques ordonnées : la réplicabilité, la répétabilité, la compréhension, la réutilisabilité, l'ouverture et l'adaptabilité.

Notre problématique de rendre les processus d'analyse capitalisables s'est ensuite décomposée en trois points : comment partager et combiner des processus d'analyse mis en œuvre dans différents outils d'analyse ? comment permettre de réexploiter un processus d'analyse existant pour répondre à un autre besoin d'analyse ? et comment assister les différents acteurs lors de l'élaboration et de l'exploitation de processus d'analyse ?

Pour répondre au premier point, nous avons proposé l'approche **CAPTEN-MANTA**. Cette approche consiste à **affranchir les processus d'analyse des contraintes techniques auxquelles ils sont confrontés**. Pour ce faire, nous avons proposé les méta-modèles **CAPTEN-ALLELE** qui le permettent, et les avons mis en œuvre dans le prototype **CAPTEN-APE**. Il s'agit de décrire les **concepts** mis en œuvre dans les différentes étapes d'un processus d'analyse, plutôt que d'utiliser directement leurs instances techniques. Nous avons donc choisi une approche en rupture avec les travaux actuels qui consistent à permettre à un processus d'analyse d'être directement exécutable afin d'obtenir des résultats : nous nous établissons dans un cadre plus abstrait, propice à la généricité.

Nous proposons d'éliciter en *variables indépendantes* les concepts véhiculés par les données dans les traces pour les affranchir de leur contexte technique particulier. Pour cela, nous proposons conjointement un méta-modèle de variables et de listes, qui sont des conteneurs destinés à enrichir sémantiquement les variables.

Nous proposons également la notion d'*opérateur indépendant*, qui représente le concept d'opération commun entre des opérateurs similaires mais implémentés dans des outils d'analyse différents. Le méta-modèle associé nous permet d'atteindre l'indépendance technique des opérateurs en ne conservant que

les prérequis fondamentaux à leur utilisation – en tant que concepts d'opération – et leur archétype comportemental. Du reste, nous définissons le comportement de ces opérateurs indépendants à l'aide de seulement quelques règles qui, une fois combinées entre elles, permettent de décrire à un haut niveau la totalité des effets observables des opérateurs implémentés sur les traces.

En outre, nous avons vu qu'un processus d'analyse, une fois constitué uniquement d'opérateurs indépendants et de variables qui le sont tout autant, devient également indépendant des contraintes techniques. Nous en proposons un méta-modèle faisant intervenir conjointement opérateurs indépendants et variables. Décrire un processus d'analyse selon notre approche revient à décrire le modèle générique à suivre pour obtenir des connaissances spécifiques. Pour les exploiter concrètement, il convient de les implémenter dans des outils d'analyse qui possèdent les instances des opérateurs indépendants nécessaires. Pour assister l'implémentation et la mise en relation entre opérateur indépendant et opérateurs implémentés dans les outils, chaque opérateur indépendant indique quels opérateurs implémentés véhiculent concrètement le concept d'opération qu'il représente.

Aussi, en proposant une indépendance technique, nous couvrons les propriétés de réplicabilité et de reproductibilité nécessaires à la capitalisation des processus d'analyse de traces d'apprentissage. Cette proposition constitue ainsi le socle théorique de nos travaux sur la capitalisation des processus d'analyse de traces. En outre, nous renforçons aussi le partage de ces processus d'analyse puisqu'ils sont assimilables à des modèles génériques d'analyse.

Pour répondre au deuxième point, à savoir comment permettre de réexploiter un processus d'analyse existant pour répondre à un autre besoin d'analyse, nous avons proposé l'approche **CAPTEN-ATOM**. Cette approche consiste à **narrer les processus d'analyse de traces d'apprentissage**. Pour ce faire, nous avons proposé une ontologie et les concepts sous-jacents (**CAPTEN-ONION**), que nous avons mis en œuvre dans **CAPTEN-TORTOISE**. Cette approche est une suite directe à notre proposition précédente concernant l'indépendance technique des processus d'analyse.

Par conséquent, il s'agit toujours de décrire les concepts qui sont mis en œuvre dans l'analyse, et donc de favoriser une représentation abstraite des processus d'analyse de traces. Toutefois, à la différence de nos travaux sur l'indépendance technique, nous avons opté pour une approche sémantique forte, nous permettant d'enrichir le processus d'analyse avec des informations qui lui sont directement intégrées et structurées. Cette approche est donc doublement en rupture avec les travaux actuels qui d'une part permettent à un processus d'analyse d'être directement exécutable ; et qui d'autre part ne permettent pas d'intégrer l'information efficacement dans le processus d'analyse pour l'exploiter ensuite – ces informations ne concernant pas toutes les étapes de l'élaboration ainsi que ses acteurs. Nous établissons un cadre propice à rendre les processus d'analyse comme des artefacts autosuffisants pour les différents acteurs de l'analyse.

Nous avons élaboré notre ontologie pour répondre spécifiquement à chacune des propriétés de la capitalisation. Pour permettre la **réplicabilité** des processus d'analyse, nous proposons le concept d'*opérateur narré* et celui de *processus d'analyse narré*. Un opérateur narré représente le concept d'opération commun entre des opérateurs similaires, où sa sémantique est non ambigüe, et où différentes informations y sont intégrées et structurées. Nous définissons le comportement de ces opérateurs sur les variables à l'aide de différents patrons, eux aussi sémantiquement définis. Ayant préalablement montré qu'un processus d'analyse peut être indépendant techniquement si l'ensemble de ses composants l'est, un processus d'analyse narré n'y déroge pas. Les opérations qui y sont réalisées sont représentées par les opérateurs narrés – inclus dans des étapes. À un processus d'analyse narré, nous donnons la possibilité d'intégrer également de l'information structurée.

Pour permettre la **répétabilité** d'un processus d'analyse, nous proposons de conserver l'abstraction opérée sur les traces, en ne conservant que les variables (*i.e.* les concepts de données). Toutefois, nous donnons la possibilité de représenter les relations – explicites et implicites – qui peuvent exister entre ces variables : cela contribue à définir un *graphe de variables* qui est potentiellement porteur d'informations supplémentaires par rapport aux traces sources, qu'il est possible d'exploiter.

Pour favoriser la **compréhension** des processus d'analyse narrés, nous proposons le concept d'*élément narratif*. Ce sont ces éléments qui permettent d'intégrer l'information d'une manière structurée. Il s'agit

d'un type d'information sémantisée, mis en relation avec n'importe quels autres éléments, y compris d'autres éléments narratifs, peuplé d'un contenu structuré.

La **réutilisation** est respectée par la définition de trois éléments distincts. Tout d'abord, le concept d'*étape de l'analyse*. Une étape capture la noèse opérée par l'analyste au niveau atomique de l'analyse : l'intention de l'analyste d'effectuer une opération spécifique sur des données spécifiques. Elle est donc constituée d'un opérateur narré, d'un graphe de variables d'entrée et d'un graphe de sortie. Cela nous permet d'avoir un niveau de granularité supplémentaire entre le processus d'analyse et les opérateurs utilisés, et donc de pouvoir décrire ces intentions – notamment *via* des éléments narratifs. Nous donnons aussi la possibilité de décrire quels besoins sont répondus, grâce à l'introduction du *concept de connaissance*, qui nous permet d'identifier sémantiquement les variables produites pertinentes. Enfin, nous conservons dans l'ontologie la possibilité d'indiquer les outils d'analyse qui implémentent les opérateurs narrés.

L'**ouverture** est favorisée par l'adoption d'un *vocabulaire contrôlé*, qui se traduit par un ensemble de classes et de propriétés sémantiques définies. C'est ce vocabulaire qui est utilisé pour décrire les différents éléments instanciés dans l'ontologie. De plus, dans une optique d'interopérabilité, nous réutilisons des termes issus de différents travaux, comme xAPI ou wf4ever, dans l'objectif à terme de créer un effort commun.

Enfin, nous permettons aux processus d'analyse narrés d'être **adaptables** en apportant une attention particulière à la modélisation du contexte de l'analyse. Pour cela, nous définissons le concept de *contexte* dans l'ontologie. Le contexte de l'analyse est ensuite utilisé pour conditionner l'impact des opérateurs narrés dans le processus d'analyse, et documenter différents éléments en même temps que les éléments narratifs.

Ce nouveau paradigme de représentation des processus d'analyse permet ainsi de les représenter d'une manière unifiée, tant leur structure que leurs informations. Il est aussi possible de les documenter d'après les différentes étapes de leur cycle d'élaboration. Aussi, cela permet de fournir différents niveaux de lecture du processus pour les différents acteurs du cycle. Il en résulte également des possibilités d'assistances nouvelles et importantes.

Enfin, pour répondre au troisième point concernant l'**assistance** des différents acteurs de l'analyse, symbolisé par notre approche **CAPTEN-AERIS**, nous avons proposé une **recherche intelligente des processus d'analyse** utilisant la narration précédemment mise en place. L'objectif est d'exploiter la sémantique inhérente aux processus d'analyse narrés, et à leurs éléments, pour permettre une meilleure interprétation par la machine. Nous en avons proposé les concepts et l'algorithmie (**CAPTEN-FRUIT**), que nous avons mis en œuvre dans **CAPTEN-SEED**.

Nous proposons d'effectuer la recherche non plus en fonction de requêtes simples, ni de textes libres, mais d'après l'expression d'un besoin d'analyse. Pour cela, nous proposons de définir un *besoin d'analyse* comme un ensemble de *dimensions* finies, qui définissent une propriété spécifique du besoin, comme le contexte pédagogique, ou encore l'objectif.

Pour décrire ces dimensions, nous introduisons la notion de *token*, qui sont des termes qui statuent d'un point spécifique de la dimension en question. Ces termes proviennent du vocabulaire contrôlé défini avec notre approche narrative, et peuvent donc être des classes ou des relations sémantiques. Cela nous permet d'exploiter les propriétés sémantiques définies dans notre ontologie avec les tokens lors de la recherche, et ainsi d'affiner la recherche et les candidats potentiels.

Pour réaliser la recherche, nous projetons dans notre ontologie chacune des dimensions du besoin. Chaque projection est régie par un ensemble de règles pour requêter l'ontologie, que nous appelons des *patrons de recherche*. Ce sont des heuristiques de haut niveau pour l'utilisateur, qui définissent quels éléments de l'ontologie sont censés être pertinents pour la dimension associée. Lors de cette recherche, nous faisons également intervenir la notion de *similarité* entre les termes, pour étendre l'espace de recherche de manière intuitive pour l'utilisateur. Si un terme est absent, alors l'utilisateur peut choisir d'utiliser des termes qui lui sont similaires.

De plus, dans l'idée de placer l'utilisateur au centre du processus de recherche, nous lui donnons la possibilité de préciser l'importance de certaines dimensions, ou de certains termes, toujours avec un niveau d'abstraction élevé. Nous utilisons pour cela des *modificateurs flous*. De plus, nous avons introduit la possibilité de *relaxer* des contraintes sur la définition du besoin lors de la recherche, afin d'étendre l'espace de recherche de manière contrôlée.

Enfin, nous proposons de construire une liste de résultats au besoin d'analyse, enrichis d'un *score de pertinence*. Et nous proposons, pour chacun des résultats, une explication des raisons qui ont conduit le système à attribuer le score de pertinence associé à l'utilisateur, en utilisant un mécanisme de *backtracking*. L'objectif ici est d'assister l'utilisateur dans l'identification des candidats potentiels.

Pour conclure, les trois approches que nous avons proposées, à savoir **CAPTEN-MANTA**, **CAPTEN-ATOM** et **CAPTEN-AERIS**, contribuent à un cadre conceptuel cohérent qui permet de capitaliser les processus d'analyse de traces d'apprentissage et qui permet de mettre en place de nouvelles assistances pour les différents acteurs de l'analyse. Les résultats expérimentaux appuient cette conclusion, et nous incitent à en conduire de nouvelles avec un nombre de personnes plus important, notamment pour évaluer plus en détail la propriété d'ouverture.

Nous terminons cette thèse en abordant, dans le chapitre suivant, différentes perspectives que nous pensons importantes à considérer pour apprécier tout l'intérêt d'une approche narrative des processus d'analyse de traces d'apprentissage.

Perspectives

Sommaire

Section	**Développer une banque commune de processus d'analyse narrés**	**207**
Section	**Détecter les éléments critiques dans les processus d'analyse narrés**	**208**
Section	**Adapter automatiquement les processus d'analyse de traces**	**208**
Section	**Instancier automatiquement dans les outils d'analyse**	**209**
Section	**Cycle de vie des processus d'analyse narrés exploitant la sémantique**	**210**
Section	**Narration et science ouverte**	**210**
Section	**Tendre vers un raisonnement à partir de cas**	**211**

Développer une banque commune de processus d'analyse narrés

En s'inspirant des travaux communautaires comme ^{my}Experiment (C. A. GOBLE et al., 2010) ou Tigris (LEARNSPHERE, 2018[d]), la première perspective pour nos travaux serait la mise en place d'une banque commune de processus d'analyse narrés. Cela permettrait de rassembler la communauté des Learning Analytics et de lui proposer un endroit où les processus d'analyse destinés à l'analyse de l'apprentissage seraient regroupés, décrits de manière compréhensible, adaptables et réutilisables.

Conséquemment, il est naturel de prévoir également un espace communautaire servant d'interface à cette banque commune. Cet espace permettrait de visualiser les différents aspects proposés par notre approche narrative, comme les processus d'analyse narrés, mais aussi les opérateurs narrés, les graphes de variables, ou encore le vocabulaire contrôlé. Nous pensons d'ailleurs qu'il s'agit ici d'une opportunité pour la communauté de développer un vocabulaire commun pour parler des analyses, des variables, des éléments narratifs et aussi des paramétrages. Actuellement, la pluridisciplinarité du domaine des EIAH et des acteurs entraînent des difficultés pour établir un consensus, comme cela s'observe entre les deux standards Caliper et xAPI.

De plus, un tel espace permettrait aux utilisateurs de proposer des retours d'utilisation des processus d'analyse narrés adaptés et réutilisés. Cela serait notamment très utile pour modifier la narration des processus d'analyse de traces en fonction de différents paramètres.

Notons que le prototype **CAPTEN-TORTOISE** mettant en œuvre la narration a été développé avec l'objectif initial d'être la *front-end* de ce genre de dispositif communautaire. Toutefois, nos travaux peuvent être intégrés dans des dispositifs déjà existants et ainsi les compléter par la modélisation des analyses. Nous pensons notamment à la plateforme Hubble, destinée à stocker la description des analyses sous forme de texte libre (PROJET HUBBLE, 2018). Cette description textuelle pourrait alors être présentée conjointement avec la description narrative, qui formalise l'analyse. Nous pouvons même envisager de lier ces deux descriptions grâce aux éléments narratifs.

Détecter les éléments critiques dans les processus d'analyse narrés

Dans cette section, nous nous intéressons à un autre type d'assistance sur lequel nous avons travaillé et qu'il serait intéressant de mettre en place. Il s'agit de découvrir automatiquement des éléments critiques en lien avec l'analyse. Nous considérons un élément d'un processus d'analyse comme critique si, lorsque le contexte du processus change, il n'est plus pertinent et occasionne des erreurs (*e.g.* d'interprétation). Comme on le pressent, c'est principalement le paramétrage des opérateurs qui va être concerné.

Pour remédier à ces effets de bords qui surviennent lors de l'adaptation des processus d'analyse, nous prévoyons de nous appuyer sur les éléments narratifs qui sont utilisés lors de la description du processus d'analyse narré, ainsi que sur les retours utilisateurs. En effet, dans le cadre d'une narration correctement conduite, les différents éléments constitutifs de l'analyse sont mis en relation les uns les autres. Par exemple, le paramétrage d'un opérateur peut dépendre d'une variable d'un graphe de variables. Si ce dernier est contextualisé, alors ce contexte se répercute sur la configuration même de l'opérateur.

Dès lors, nous pensons qu'il est possible d'identifier une classe d'éléments narratifs qui contraint l'application des éléments et qui exprime dans quelle situation il est pertinent de les utiliser. C'est par exemple le cas du contexte, comme nous l'avons vu, mais aussi des hypothèses, entre autres. Cette classe définie, il devient possible d'analyser sémantiquement chaque processus pour vérifier s'il possède des éléments critiques. Pour cela, il faut, à l'instar de la recherche intelligente présentée dans le Chapitre 9, définir un ensemble de requêtes dédiées pour vérifier l'existence de ces éléments narratifs, et en vérifier le contenu. De plus, les éléments identifiés comme critiques pourront également être indiqués à l'utilisateur.

Enfin, par les retours utilisateurs, nous prévoyons d'affiner le mécanisme de détection, notamment *via* un apprentissage par renforcement comme celle du Q-learning. Une telle approche, brièvement, ne nécessite pas de modèle initial de l'environnement, et consiste en un ensemble d'états et d'actions ayant un coût. Aussi, chaque retour utilisateur pourrait modifier le coefficient de pondération associé aux actions dont le but est d'identifier les éléments critiques, et les règles associées à ces détections.

Il serait en effet intéressant de récupérer les retours d'expérience utilisateurs lorsque l'analyse a été réutilisée dans d'autres contextes, et d'affiner les requêtes en fonction des éléments qui ont effectivement été critiques ou non. Grâce à cette correspondance évolutive, lors de la réutilisation d'un processus d'analyse narré, il sera alors possible d'assister l'analyste en lui indiquant les éléments auxquels il doit porter attention. De plus, en combinant cette correspondance avec la recherche intelligente, il est possible d'indiquer directement les éléments des solutions qui ne conviennent pas dans le contexte du besoin recherché.

Adapter automatiquement les processus d'analyse de traces

Nous envisageons également une autre assistance, celle de l'adaptation automatique des processus d'analyse narrés, en fonction d'un besoin d'analyse exprimé par l'utilisateur. Pour cela nous prévoyons de nous appuyer à la fois sur la détection des éléments critiques, et la recherche intelligente. En effet, afin d'être capable d'adapter les processus automatiquement, il est impératif que la machine puisse identifier de manière autonome les éléments à modifier, et trouver des substituts qui conviennent. Il nous semble aussi pertinent de ne chercher à adapter que les processus qui correspondent le plus au besoin recherché par l'utilisateur.

Par conséquent, la recherche intelligente est une première étape intéressante, puisqu'elle nous permet de récupérer une liste de résultats, organisée par score de pertinence : ceux avec le score le plus

important sont supposés répondre le mieux au besoin d'analyse exprimé par l'utilisateur. C'est sur cette liste de résultats que l'on peut appliquer la détection d'éléments critiques vue dans la perspective précédente. Cela nous permettrait de vérifier si chaque processus d'analyse peut s'appliquer convenablement au besoin exprimé par l'utilisateur. Si la détection nous rapporte des éléments jugés critiques, alors, pour chaque élément critique, une alternative est recherchée, toujours avec la recherche intelligente, contextualisée d'après le besoin utilisateur.

De plus, il est possible avec cette approche de générer la trace du raisonnement effectué par le système. De cette manière, nous prévoyons d'expliquer à l'utilisateur les inférences effectuées par la machine et de le laisser juge de la pertinence de l'adaptation réalisée. Mais surtout, cela nous permettra de demander à l'utilisateur si les propositions d'adaptation automatique lui semblent non pertinentes, et lesquelles. Ainsi, grâce aux réponses de l'analyste, la détection d'informations critiques et les mécanismes d'adaptation pourront être affinés, et le processus d'adaptation amélioré.

Instancier automatiquement dans les outils d'analyse

Nos approches **CAPTEN-MANTA** et **CAPTEN-ATOM** s'effectuent dans un cadre plus abstrait que les approches traditionnelles. Conscients de cela, nous avons dès le début intégré à nos modélisations des pointeurs d'instanciation vers les différents outils d'analyse, et les opérateurs correspondants. De cette manière, il est possible de produire à partir d'un processus d'analyse narré sa notice d'implémentation pour différents outils d'analyse. Chaque étape narrée peut en effet être mise en correspondance avec un opérateur implémenté dans un outil d'analyse, avec les paramétrages à adopter.

En parallèle, il serait aussi intéressant d'envisager de proposer un mécanisme d'instanciation automatique, basé sur ces informations intégrées à la modélisation des processus d'analyse. Un tel mécanisme permettrait alors aux différents acteurs de jouer les analyses narrées directement dans les outils d'analyse de leur choix. Concrètement, il s'agit là d'un atout non négligeable pour les acteurs, puisqu'il devient possible d'avoir un retour immédiat sur le fait que les processus d'analyse réutilisés conviennent bien à leur situation. En sus, cela leur permettrait d'évaluer directement l'effet des configurations appliquées aux opérations narrées et aux étapes, voire même dans le cadre d'un raisonnement à partir de cas [5] de procéder à un retour d'expérience directe pour aiguiller le système. Enfin, cela constituerait également une assistance majeure puisque l'instanciation serait déléguée au système, rendant accessible la mise en œuvre des analyses à des acteurs non experts de l'analyse – et ainsi répondre rapidement à des besoins d'analyse.

La première des nécessités pour permettre cette instanciation automatique est de rendre bijectif chaque opérateur narré avec les opérateurs de chaque outil d'analyse qu'il représente. Cela est nécessaire afin d'identifier quel(s) opérateur(s) utiliser. Pour cela, il est nécessaire d'embarquer les spécificités techniques propres à chacun de ces outils dans ce mécanisme d'instanciation. Ensuite, il est nécessaire de procéder à une mise en correspondance de chaque variable du graphe de variables initiales du processus d'analyse narré avec les traces à disposition dans l'outil d'analyse. Là aussi, il est nécessaire d'embarquer comment l'outil d'analyse en question représente et gère ces traces. Il en va de même pour toutes les configurations des opérateurs, et de leurs effets.

Enfin, il est nécessaire de piloter l'outil d'analyse de manière automatique, ce qui peut se faire par exemple *via* des interfaces de type REST. De cette manière, les utilisateurs pourraient alors accéder à un espace communautaire, décider de réutiliser une analyse et l'exécuter directement dans l'outil de leur choix. De cette manière, notre approche narrative jouerait en plus le rôle d'un langage intermédiaire entre les différents outils d'analyse, à l'instar de PMML (Data Mining Group, 2018[j]) pour les modèles prédictifs. Ainsi, cela permettrait aux processus d'être interopérables.

Bien que dans le cadre de cette thèse nous n'ayons pas proposé ces travaux, nous pensons qu'ils sont importants, notamment pour contrebalancer plus encore l'effort nécessaire à la narration des processus d'analyse. Toutefois, il s'agit d'un énorme travail, à la fois de recherche et d'ingénierie, qui est entièrement dépendant des outils d'analyse, et de leurs évolutions.

5. Voir la perspective *Tendre vers un raisonnement à partir de cas* pour plus d'explication sur ce dernier.

Cycle de vie des processus d'analyse narrés exploitant la sémantique

Une autre perspective que permet notre approche narrative est le versionnage des processus d'analyse narrés. Lorsqu'un utilisateur a adapté, puis réutilisé, un processus d'analyse narré, nous pouvons supposer qu'il aimerait rapporter ce nouveau processus dans la base commune des processus d'analyse narrés et faire part de son retour d'expérience.

Actuellement, le versionnage de processus d'analyse est relativement problématique, notamment lorsqu'il s'agit de processus qui sont amenés à évoluer rapidement, et sans réelle supervision, comme cela peut être le cas dans des approches communautaires (FREIRE et al., 2006). Néanmoins, des travaux scientifiques, notamment dans le domaine des workflows, proposent déjà des pistes intéressantes, comme ceux de Freire & *al.* (FREIRE et al., 2006) ou ceux de Callahan & *al.* (CALLAHAN et al., 2006).

Ces travaux proposent de tenir compte de l'évolution d'un workflow grâce à un arbre de dépendance. Chaque nœud de l'arbre représente alors une version spécifique du workflow : un nœud fils est obligatoirement issu de la version du workflow représentée par le nœud père. Cela permet d'explorer l'évolution du processus d'analyse.

En nous inspirant de ces travaux, nous pensons que notre approche narrative peut enrichir cette approche de versionnage. En effet, dans ces travaux, seuls les nœuds sont labélisés et sont supposés traduire la différence entre deux workflows. Avec notre approche narrative, nous pouvons reproduire cette relation et constituer l'arbre de dépendance. L'avantage est que nous utilisons des termes sémantiquement définis pour représenter ces labels : de cette manière, l'on peut envisager d'utiliser cette notion de version jusque dans les assistances. En plus de cela, il est aussi possible d'intégrer des informations précises aux différents éléments du processus qui ont été modifiés, ainsi que les raisons de ces modifications. Cela contribuerait de ce fait à renforcer la contextualisation des différentes versions d'un processus d'analyse.

Enfin, nous nous demandons si, grâce à un tel versionnage, cela permettrait de faire poindre certaines relations entre techniques d'analyse, contexte pédagogique et besoin d'analyse. Cela favoriserait alors l'émergence d'études dédiées à ces spécificités et permettrait peut-être de renforcer certaines pratiques des Learning Analytics (S. DAWSON et al., 2019).

Narration et science ouverte

Nous pensons qu'il serait également intéressant d'étudier comment les processus d'analyse narrés, qui traduisent la méthodologie réalisée pour obtenir des connaissances précises, pourraient être utilisés pour renforcer la démarche scientifique dans notre discipline, et apporter plus de crédibilité et de reproductibilité scientifique aux différents travaux.

Il serait approprié dans tous les domaines, lors des étapes de reviews par les pairs des travaux de recherche, de fournir un accès direct aux processus d'analyse associés. Cela permettrait alors de fournir un support supplémentaire lors de l'évaluation des travaux de recherche, principalement en favorisant la reproductibilité de l'ensemble du processus d'analyse et l'échange scientifique (*e.g.* discuter les hypothèses sous-tendant l'analyse) – et ainsi peut-être modifier certaines habitudes de publications. Pour cela, il serait intéressant soit de fournir directement les processus d'analyse narrés (*e.g.* exportés et joins, par exemple, dans une archive), soit de fournir un lien vers eux dans l'espace communautaire. Il serait là encore intéressant d'intégrer les retours d'évaluation des pairs directement dans des éléments narratifs dédiés – qui pourront selon la nature de ces éléments être mis en relation avec d'autres éléments du framework.

Pour contribuer à renforcer la traçabilité des analyses, ainsi que leur pérennité, l'on peut également envisager d'inclure directement les processus d'analyse narrés dans les publications scientifiques. Cela peut se faire par exemple par l'entremise d'un code-barres bidimensionel (*e.g.* Code QR) encodant

une URL vers l'espace communautaire, ou encore directement vers la banque commune de processus d'analyse narrés. *De facto*, cela assiérait une visibilité et une crédibilité scientifique accrues dans notre discipline.

En outre, tout en prenant en compte les différentes perspectives vues jusqu'à présent, avoir un lien direct vers ces analyses narrées dans les travaux de recherche éveille une perspective stimulante. Il devient en effet possible d'évaluer ces travaux de recherche dans d'autres contextes pédagogiques, et de faire part des retours sous forme d'éléments narratifs et de cycle de vie : les publications ne sont plus figées dans le temps, et évoluent ainsi de concert avec la communauté, ses besoins et ses efforts.

Tendre vers un raisonnement à partir de cas

D'autres perspectives à nos travaux émergent lorsque l'on observe la forte similarité entre le raisonnement à partir de cas (KOLODNER, 1992) (RàPC) et la mise en œuvre conjointe du mécanisme de recherche d'analyses présenté dans cette thèse, et de la détection d'éléments critiques et de l'adaptation automatique vues en perspective. Brièvement, un processus de raisonnement à partir de cas tente de répondre à un problème donné – dit source – en consultant sa base de cas (problèmes résolus). Pour cela, ce processus est composé de quatre étapes principales : 1) une étape de recherche des cas similaires au problème source ; 2) une étape de réutilisation qui consiste à faire correspondre un cas trouvé dans la base – dit cible – de cas au problème source ; 3) une étape de révision où, si la solution n'a pas fonctionné, il est possible de corriger la solution, ou d'indiquer les éléments ou les raisons de cet échec ; 4) une étape d'apprentissage, où le système enregistre le nouveau cas constitué du problème source et de sa solution, ou l'échec.

Nous voyons une analogie forte entre ce cycle du raisonnement à partir de cas, et l'enchaînement des assistances que nous avons préalablement évoquées. Ainsi, le problème source du RàPC peut être assimilé au besoin d'analyse décrit par l'utilisateur. L'étape de recherche du cycle peut, quant à elle, être assimilée à notre recherche intelligente, proposée dans cette thèse. Concernant l'étape de réutilisation où des modifications peuvent être effectuées pour faire correspondre la solution, nous voyons là l'appel conjoint des deux assistances précédemment évoquées : la détection des éléments critiques, puis leur adaptation, pour proposer à l'utilisateur des processus d'analyse narrés pertinents pour son besoin.

S'ensuit alors, après la réutilisation du processus d'analyse narré, l'étape de révision. Dans notre cas, c'est l'utilisateur qui, par ces retours d'utilisation, pourra indiquer les éléments du processus qui n'ont pas fonctionné. De plus, il est aussi envisageable de demander à l'utilisateur si les éléments critiques ont été correctement détectés, et si l'adaptation a correctement été réalisée. En fonction des réponses, l'on peut envisager une modification des requêtes [6].

Enfin, suite au retour de l'utilisateur, le nouveau processus d'analyse est également narré, et ajouté à la base des processus d'analyse narrés. Les spécificités par rapport au processus d'analyse narré d'origine peuvent également être indiquées, notamment grâce aux éléments narratifs.

Une perspective qui résulte de cette mise en correspondance avec le RàPC serait de fournir aux différents acteurs de l'analyse un mécanisme dédié, personnalisé et personnalisable, pour les assister dans leur tâche de réutilisation de l'existant. Par exemple, le mécanisme de réutilisation (cf. étape 2 du RàPC) pourrait être adapté en fonction de l'expertise de l'utilisateur. De plus, cela servirait à l'enrichissement de la base des processus d'analyse narrés d'une manière manuelle, et ce, de manière contrôlée. En effet, les retours utilisateur seraient directement enregistrés et permettraient de modifier plus que les coefficients de pondération pour la détection d'information critique [7] : les politiques d'adaptation du système pour les processus d'analyse pourraient ainsi être modélisées sous forme de règles, et pondérées suivant ces retours.

6. Par exemple, en associant et en faisant moduler un coefficient de pondération associée à chacune des requêtes.
7. Il s'agit du coefficient suggéré dans la perspective *Détecter les éléments critiques dans les processus d'analyses narrés*, qui consiste à affiner les règles d'identification des éléments *via* les retours utilisateurs.

De surcroît, cela nous ouvre une nouvelle perspective, fortement intéressante. Il s'agit de l'enrichissement automatique de la base de processus d'analyse. Puisque le système dispose d'un ensemble de règles et de contraintes, supervisé par l'humain, concernant la détection d'éléments critiques et l'adaptation des processus d'analyse, il devient possible pour le système de générer des besoins d'analyse, et de proposer des processus d'analyse y répondant. Ces processus pourront ensuite être mis en œuvre afin d'être testés, et les retours d'expérience serviront directement au système pour affiner ses règles.

Finalement, disposer d'un tel cycle nous permet également d'envisager la proposition de solutions issues d'une approche multi-analyses : plus qu'un processus d'analyse ou un sous-ensemble d'étapes, c'est un processus d'analyse construit d'après des parties de ceux déjà existants qui serait alors proposé. Pour cela, le système identifierait dans sa base les processus d'analyse qui produisent les connaissances attendues, ainsi que ceux partant du contexte initial de l'utilisateur, et chercherait comment les combiner ensemble, tout en respectant les éléments critiques définis par l'utilisateur. Les graphes de variables de sortie et d'entrée joueraient là un rôle essentiel : ceux d'entrée indiquant quelles variables sont nécessaires pour utiliser des étapes ou suites d'étapes, ceux de sortie représentant les variables disponibles à un instant donné. Néanmoins, nous pensons que cette approche multi-analyses serait confrontée à un problème combinatoire important, qu'il sera impérieux de résoudre pour qu'elle soit concrètement applicable.

Pour conclure, nous pensons que notre travail dans son ensemble conviendrait à d'autres disciplines amenées à traiter des données en réalisant des pipelines d'opérations. Nous pensons par exemple au domaine de l'image informatique, où des artefacts numériques (*e.g.* rivières, nuages, animations) sont créés à partir de données. Offrir une banque d'analyses apportant une modélisation abstraite de ces pipelines, les regroupant et permettant de les combiner servirait ces domaines, au même titre que le nôtre (*e.g.* consultation de l'existant, émergence d'un vocabulaire commun, réutilisation facilitée).

> *Soyez résolus de ne servir plus, et vous serez libres.*
>
> **— Étienne de La Boétie**
> (Discours de la servitude volontaire)

Partie VI

Bibliographie

Bibliographie

ACM (2017). *Artifact Review and Badging*. URL : http://www.acm.org/publications/policies/artifact-review-badging (visité le 22 nov. 2017) (cf. p. 86).

ADDIS, M., J. FERRIS, M. GREENWOOD et al. (2003). "Experiences with e-Science workflow specification and enactment in bioinformatics". In : *Proceedings of UK e-Science All Hands Meeting*, p. 459–466 (cf. p. 38).

ADVANCED DISTRIBUTED LEARNING (2013). *Experience API (xAPI) - TinCan*. URL : https://xapi.com/ (visité le 22 juin 2018) (cf. p. 12, 27, 42, 97, 153).

– (2018[a]). *SCORM Overview*. URL : http://adlnet.gov/scorm (visité le 21 juin 2018) (cf. p. 26).

– (2018[b]). *xAPI Deep Dive : Extensions*. URL : https://xapi.com/deep-dive-extensions/ (visité le 1er juil. 2018) (cf. p. 27).

– (2018[c]). *xAPI Specification*. URL : https://github.com/adlnet/xAPI-Spec/blob/master/xAPI-Communication.md (visité le 6 août 2018) (cf. p. 27).

– (2018[d]). *xAPI Statement Data Model represented as RDF Classes and Properties*. URL : https://github.com/adlnet/xapi-ontology (visité le 20 août 2018) (cf. p. 42).

– (2018[e]). *xAPI : Statements 101*. URL : https://xapi.com/statements-101/ (visité le 28 juin 2018) (cf. p. 13, 14).

– (2018[f]). *xAPI Vocabulary*. URL : http://xapi.vocab.pub/browse/index.html (visité le 22 août 2018) (cf. p. 42).

– (2018[g]). *xAPI Vocabulary - Improving Semantic Interoperability of Controlled Vocabularies*. URL : https://fr.slideshare.net/jhaag75/xapi-vocabulary-improving-semantic-interoperability-of-controlled-vocabularies (visité le 20 août 2018) (cf. p. 42, 43).

AGNIHOTRI, L., S. MOJARAD, N. LEWKOW et A. ESSA (2016). "Educational Data Mining with Python and Apache Spark : A Hands-on Tutorial". In : *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge*. LAK '16. Edinburgh, United Kingdom : ACM, p. 507–508 (cf. p. 16, 83, 124).

– (2019). *GitHub - Educational Data Mining with Python and Apache Spark : A Hands-on Tutorial*. URL : https://github.com/Lewkow/LAK_2016_Workshop (visité le 14 mar. 2019) (cf. p. 83).

ALTINTAS, I., C. BERKLEY, E. JAEGER et al. (2004). "Kepler : an extensible system for design and execution of scientific workflows". In : *Proceedings. 16th International Conference on Scientific and Statistical Database Management, 2004.* P. 423–424 (cf. p. 38, 39).

APEREO FOUNDATION (2016). *Open Learning Analytics Platform*. URL : https://www.apereo.org/sites/default/files/projects/Brochures/Apereo%20Analytics%20Briefing%2026Apr16.pdf (visité le 25 juin 2018) (cf. p. 22, 31, 88, 131).

– (2018). *Learning Analytics Processor*. URL : http://apereo-learning-analytics-initiative.github.io/LearningAnalyticsProcessor/ (visité le 25 juin 2018) (cf. p. 22).

BAADER, F., D. CALVANESE, D. L. MCGUINNESS, D. NARDI et P. F. PATEL-SCHNEIDER (2003). *The Description Logic Handbook : Theory, Implementation, and Applications*. New York, NY, USA : Cambridge University Press (cf. p. 59).

BAKER, B. M. (2007). “A conceptual framework for making knowledge actionable through capital formation”. Thèse de doct. University of Maryland University College (cf. p. 14–16).

BAKER, R. S. J. D. et K. YACEF (2009). “The State of Educational Data Mining in 2009 : A Review and Future Visions”. In : *Journal of Educational Data Mining* 1.1, p. 3–16 (cf. p. 12, 16, 17, 28, 67, 111, 150).

BAKER, R. S. J. D., A. T. CORBETT, K. R. KOEDINGER et A. Z. WAGNER (2004). “Off-task Behavior in the Cognitive Tutor Classroom : When Students "Game the System"”. In : *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’04. Vienna, Austria : ACM, p. 383–390 (cf. p. 2).

BÁNÁTI, A., P. KACSUK et M. KOZLOVSZKY (2015). “Minimal sufficient information about the scientific workflows to create reproducible experiment”. In : *2015 IEEE 19th International Conference on Intelligent Engineering Systems (INES)*, p. 189–194 (cf. p. 29, 38, 55, 86, 87).

– (2016). “Investigation of the descriptors to make the scientific workflows reproducible”. In : *2016 IEEE 17th International Symposium on Computational Intelligence and Informatics (CINTI)*, p. 000129–000134 (cf. p. 38).

BANDROWSKI, A., R. BRINKMAN, M. BROCHHAUSEN et al. (2016). “The Ontology for Biomedical Investigations”. In : *PLOS ONE* 11.4, p. 1–19 (cf. p. 53).

– (2018). *Ontology for Biomedical Investigations*. URL : http://obi-ontology.org/ (visité le 5 sept. 2018) (cf. p. 53).

BECHHOFER, S., I. BUCHAN, D. D. ROURE et al. (2013). “Why linked data is not enough for scientists”. In : t. 29. 2. Special section : Recent advances in e-Science, p. 599–611 (cf. p. 42, 50, 53, 55).

BEEK, M. van den (2018). *Workflow version managment*. URL : https://github.com/galaxyproject/galaxy/issues/6410 (visité le 27 sept. 2018) (cf. p. 69).

BELHAJJAME, K., G. KLYNE, D. GARIJO VERDEJO et al. (2013a). *Wf4Ever Research Object Model 1.0*. URL : http://wf4ever.github.io/ro/ (visité le 27 juin 2018) (cf. p. 54, 177).

– (2013b). *Wf4Ever Research Object Model 1.0 - Workflow execution provenance*. URL : http://wf4ever.github.io/ro/#wfprov (visité le 27 août 2018) (cf. p. 48, 49).

– (2013c). *Wf4Ever Research Object Model 1.0 - Workflow provenance*. URL : http://wf4ever.github.io/ro/#wfdesc (visité le 27 août 2018) (cf. p. 45, 46, 154).

– (2018). *Wf4Ever Ontologies*. URL : http://wf4ever.github.io/ (visité le 14 sept. 2018) (cf. p. 45–48, 57, 110).

BELHAJJAME, K., O. CORCHO, D. GARIJO et al. (2012a). “Workflow-Centric Research Objects : First Class Citizens in Scholarly Discourse”. In : *Proceedings of the Second International Conference on the Future of Scholarly Communication and Scientific Publishing Sepublica 2012*, p. 1–12 (cf. p. 53–55).

BELHAJJAME, K., M. ROOS, E. GARCIA-CUESTA et al. (2012b). “Why Workflows Break — Understanding and Combating Decay in Taverna Workflows”. In : *Proceedings of the 8th International Conference on E-Science*. Washington, DC, USA : IEEE Computer Society, p. 1–9 (cf. p. 29, 39, 42, 48, 53, 55, 57, 112, 118).

BELHAJJAME, K., J. ZHAO, D. GARIJO et al. (2013d). “A Workflow PROV-corpus Based on Taverna and Wings”. In : *Proceedings of the Joint EDBT/ICDT 2013 Workshops*. EDBT ’13. Genoa, Italy : ACM, p. 331–332 (cf. p. 48, 55–57).

BELHAJJAME, K., J. ZHAO, D. GARIJO et al. (2015). “Using a suite of ontologies for preserving workflow-centric research objects”. In : *Journal of Web Semantics* 32, p. 16–42 (cf. p. 45, 50, 53, 56–58, 85, 86, 88, 94, 122).

BERLAND, M., R. S. J. D. BAKER et P. BLIKSTEIN (2014). “Educational data mining and learning analytics : Applications to constructionist research”. In : *Technology, Knowledge and Learning* 19.1, p. 205–220 (cf. p. 17).

BERTHOLD, M. R., N. CEBRON, F. DILL et al. (2009). “KNIME - the Konstanz Information Miner : Version 2.0 and Beyond”. In : *SIGKDD Explor. Newsl.* 11.1, p. 26–31 (cf. p. 160).

BIENKOWSKI, M., M. FENG et M. BARBARA (2014). “Enhancing teaching and learning through educational data mining and learning analytics : An issue brief”. In : *Proceedings of Conference on Advanced Technology for Education*, p. 1–60 (cf. p. 12, 13, 16, 42).

Bobillo, F. et U. Straccia (2016). “The fuzzy ontology reasoner fuzzyDL”. In : *Knowledge-Based Systems* 95, p. 12–34 (cf. p. 61, 62).

Bohl, O., J. Scheuhase, R. Sengler et U. Winand (2002). “The sharable content object reference model (SCORM) - a critical review”. In : t. 2, p. 950–951 (cf. p. 26).

Bouhineau, D., S. Lalle, V. Luengo et al. (2013a). “Share data treatment and analysis processes inTechnology enhanced learning”. In : *Workshop Data Analysis and Interpretation for Learning Environments*. Autrans, France (cf. p. 28, 80, 81).

Bouhineau, D., V. Luengo, N. Mandran, M. Ortega et C. Wajeman (2013b). “Conception et mise en place d’un entrepôt de traces et processus de traitement EIAH : UnderTracks”. In : *EIAH 2013 - 6e Conférence sur les Environnements Informatiques pour l’Apprentissage Humain*. Toulouse, France, p. 41–42 (cf. p. 20, 57).

Bowers, S. et B. Ludäscher (2005). “Actor-oriented design of scientific workflows”. In : *Conceptual Modeling – ER 2005*. Sous la dir. de L. Delcambre, C. Kop, H. C. Mayr, J. Mylopoulos et O. Pastor. Berlin, Heidelberg" : Springer Berlin Heidelberg", 369–384" (cf. p. 51).

Breckenridge, J. N. (1989). “Replicating cluster analysis : Method, consistency, and validity”. In : *Multivariate Behavioral Research* 24.2, p. 147–161 (cf. p. 87).

Brodaric, B. et M. Gahegan (2010). “Ontology Use for Semantic e-Science”. In : *Semant. web* 1.1,2, p. 149–153 (cf. p. 50).

Callahan, S. P., J. Freire, E. Santos et al. (2006). “Managing the Evolution of Dataflows with VisTrails”. In : *22nd International Conference on Data Engineering Workshops (ICDEW’06)*, p. 71–71 (cf. p. 210).

Champin, P.-A., Y. Prié et A. Mille (2004). “MUSETTE : a framework for Knowledge from Experience”. In : *Extraction et gestion des connaissances (EGC’2004)*. Sous la dir. de G. Hébrail, L. Lebart et J.-M. Petit. France, p. 129–134 (cf. p. 12, 22).

Chandrasekaran, B. et J. R. Josephson (1997). *The ontology of tasks and methods*. Rapp. tech. SS-97-06. AAAI Spring Symposium, p. 9–16 (cf. p. 50).

Chatti, M. A., A. L. Dyckhoff, U. Schroeder et H. Thüs (2012). “A reference model for learning analytics”. In : *International Journal of Technology Enhanced Learning* 4.5-6, p. 318–331 (cf. p. 2, 31, 33, 80, 108).

Choquet, C. et S. Iksal (2006). “Usage tracking language : a meta language for modelling tracks in tel systems.” In : *Proceedings of International Conference on Software Technologies*. INSTICC, p. 133–138 (cf. p. 21).

– (2007). “Modélisation et construction de traces d’utilisation d’une activité d’apprentissage : une approche langage pour la réingénierie d’un EIAH”. In : *Sciences et Technologies de l’Information et de la Communication pour l’Éducation et la Formation* 14, p. 419–456 (cf. p. 30, 32).

Choquet, C., V. Luengo et K. Yacef (2009). *Usage Analysis in Learning Systems*. Association for the Advancement of Computing in Education (AACE) (cf. p. 21).

Ciccarese, P., E. Wu, G. Wong et al. (2008). “The SWAN biomedical discourse ontology”. In : *Journal of Biomedical Informatics* 41.5, p. 739–751 (cf. p. 51–53, 109, 115).

– (2018). *Semantic Web Applications in Neuromedicine (SWAN) Ontology*. URL : https://www.w3.org/TR/hcls-swan/ (visité le 5 sept. 2018) (cf. p. 51, 52, 55).

Clow, D. (2013). “An overview of learning analytics”. In : *Teaching in Higher Education* 18.6, p. 683–695 (cf. p. 2, 17).

Coheris (2018). *Coheris Analytics Spad*. URL : https://www.coheris.com/produits/analytics/logiciel-data-mining/ (visité le 27 déc. 2018) (cf. p. 160).

Cooper, A. (2013). *Learning Analytics Interoperability - a survey of current literature and candidate standards*. URL : http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.650.3428&rep=rep1&type=pdf (visité le 14 mar. 2019) (cf. p. 20, 32).

Corby, O., R. Dieng-Kuntz et C. Faron Zucker (2004). “Querying the Semantic Web with Corese Search Engine”. In : *European Conference on Artificial Intelligence*. Valence, Spain (cf. p. 60, 132).

CORBY, O., R. DIENG-KUNTZ, C. FARON ZUCKER et F. GANDON (2006a). *Ontology-based Approximate Query Processing for Searching the Semantic Web with Corese*. Research Report RR-5621. INRIA, p. 36 (cf. p. 59–61, 63, 132, 137).

CORBY, O., R. DIENG-KUNTZ, F. GANDON et C. FARON-ZUCKER (2006b). "Searching the semantic web : Approximate query processing based on ontologies". In : *IEEE Intelligent Systems* 21.1, p. 20–27 (cf. p. 60, 137).

CYGANIAK, R., S. FIELD, A. GREGORY, W. HALB et J. TENNISON (2010). "Semantic Statistics : Bringing Together SDMX and SCOVO." In : *Proceedings of the WWW2010 Workshop on Linked Data on the Web* 628 (cf. p. 50, 51).

DAN, B. et M. LIBBY (2014). *FOAF Vocabulary Specification 0.99*. URL : http://xmlns.com/foaf/spec/ (visité le 20 août 2018) (cf. p. 43).

DATA MINING GROUP (2018[a]). *Data Mining Group*. URL : http://dmg.org/ (visité le 10 août 2018) (cf. p. 34).

– (2018[b]). *PFA - Motivation*. URL : http://dmg.org/pfa/docs/motivation/ (visité le 13 août 2018) (cf. p. 36).

– (2018[c]). *PFA - Specification*. URL : http://github.com/datamininggroup/pfa/releases/download/0.8.1/pfa-specification.pdf (visité le 13 août 2018) (cf. p. 36, 37).

– (2018[d]). *PFA - Tutorial 1 : Scoring Engines*. URL : http://dmg.org/pfa/docs/tutorial1/ (visité le 13 août 2018) (cf. p. 36).

– (2018[e]). *PFA - Tutorial 2 : Programming*. URL : http://dmg.org/pfa/docs/tutorial2/ (visité le 13 août 2018) (cf. p. 37).

– (2018[f]). *PMML 4.3 - Changes from PMML 4.2.1*. URL : http://dmg.org/pmml/v4-3/Changes.html (visité le 13 août 2018) (cf. p. 35).

– (2018[g]). *PMML Powered*. URL : http://dmg.org/pmml/products.html (visité le 10 août 2018) (cf. p. 34, 36).

– (2018[h]). *PMML Scope and Fields*. URL : http://dmg.org/pmml/v4-3/FieldScope.html (visité le 10 août 2018) (cf. p. 34).

– (2018[i]). *Portable Format for Analytics*. URL : http://dmg.org/pfa/index.html (visité le 13 août 2018) (cf. p. 36, 42).

– (2018[j]). *Predictive Model Markup Language (PMML)*. URL : http://dmg.org/pmml/pmml-v4-3.html (visité le 1er juil. 2018) (cf. p. 31, 34, 35, 42, 86, 209).

DATASHOP (2010). *DataShop - Import Format*. URL : https://pslcdatashop.web.cmu.edu/help?page=importFormatTd (visité le 25 juin 2018) (cf. p. 20, 27).

DAWSON, S., S. JOKSIMOVIC, O. POQUET et G. SIEMENS (2019). "Increasing the Impact of Learning Analytics". In : *Proceedings of the 9th International Conference on Learning Analytics & Knowledge*. LAK19. Tempe, AZ, USA : ACM, p. 446–455 (cf. p. 210).

DE NIES, T., F. SALLIAU, R. VERBORGH, E. MANNENS et R. VAN DE WALLE (2015). "TinCan2PROV : exposing interoperable provenance of learning processes through experience API logs". eng. In : *WWW'15 companion : proceedings of the 24th international conference on world wide web*. Florence, Italy : Association for Computing Machinery (ACM), p. 689–694 (cf. p. 43, 44).

DE ROURE, D., C. A. GOBLE, G. KLYNE et al. (2011). "Towards the Preservation of Scientific Workflows." In : *In proceedings of the 8th International Conference on Preservation of Digital Objects (iPRES 2011)*. ACM (cf. p. 39).

DE ROURE, D., C. GOBLE, S. ALEKSEJEVS et al. (2010). "Towards open science : the myExperiment approach". In : *Concurrency and Computation : Practice and Experience* 22.17, p. 2335–2353 (cf. p. 53, 57).

DE ROURE, D., C. GOBLE et R. STEVENS (2009). "The design and realisation of the Experimentmy Virtual Research Environment for social sharing of workflows". In : *Future Generation Computer Systems* 25.5, p. 561–567 (cf. p. 38).

DEELMAN, E., D. GANNON, M. SHIELDS et I. TAYLOR (2009). "Workflows and e-Science : An overview of workflow system features and capabilities". In : *Future generation computer systems* 25.5, p. 528–540 (cf. p. 37).

DEMŠAR, J., T. CURK, A. ERJAVEC et al. (2013). “Orange : Data Mining Toolbox in Python”. In : *Journal of Machine Learning Research* 14, p. 2349–2353 (cf. p. 19, 28, 56, 160).

DESMARAIS, M. C. et R. S. J. D. BAKER (2012). “A review of recent advances in learner and skill modeling in intelligent learning environments”. In : *User Modeling and User-Adapted Interaction* 22.1-2, p. 9–38 (cf. p. 25, 26).

DIMITRAKOPOULOU, A. (2004). *State of the art on Interaction and Collaboration Analysis*. (D26.1.1) EU Sixth Framework programme priority 2, Information society technology, Network of Excellence Kaleidoscope, (contract NoE IST-507838), project ICALTS : Interaction & Collaboration Analysis (cf. p. 1, 14, 97).

DOUGLAS CROCKFORD (2002). *JavaScript Object Notation*. URL : http://www.json.org/ (visité le 21 juin 2018) (cf. p. 13).

DRINGUS, L. P. (2012). “Learning analytics considered harmful.” In : *Journal of Asynchronous Learning Networks* 16.3, p. 87–100 (cf. p. 3).

DUBOIS, D. et H. PRADE (1991). “Fuzzy sets in approximate reasoning, Part 1 : Inference with possibility distributions”. In : *Fuzzy sets and systems* 40.1, p. 143–202 (cf. p. 59, 62).

DÜERST, M. et M. SUIGNARD (2005). *Internationalized Resource Identifiers (IRIs)*. URL : https://tools.ietf.org/html/rfc3987 (visité le 7 déc. 2018) (cf. p. 121).

DUVAL, E. (2011). “Attention please ! Learning analytics for visualization and recommendation”. In : *Proceedings of the 1st international conference on learning analytics and knowledge*. ACM, p. 9–17 (cf. p. 27).

DYCKHOFF, A. L., D. ZIELKE, M. BÜLTMANN, M. A. CHATTI et U. SCHROEDER (2012). “Design and implementation of a learning analytics toolkit for teachers”. In : *Journal of Educational Technology & Society* 15.3, p. 58–76 (cf. p. 3).

EDUCATIONAL DATA MINING SOCIETY (2018). *Definition of EDM*. URL : http://educationaldatamining.org/ (visité le 10 juin 2018) (cf. p. 1).

ELIAS, T. (2011). *Learning Analytics : Definitions, Processes and Potential*. URL : https://pdfs.semanticscholar.org/732e/452659685fe3950b0e515a28ce89d9c5592a.pdf (visité le 14 mar. 2019) (cf. p. 14–16).

ESCIENCE LAB RESEARCH GROUP (2018). *eScience Lab*. URL : https://www.esciencelab.org.uk/ (visité le 15 août 2018) (cf. p. 38).

FAYYAD, U., G. PIATETSKY-SHAPIRO et P. SMYTH (1996). “From data mining to knowledge discovery in databases”. In : *AI Magazine* 17.3, p. 37 (cf. p. 16, 18, 176).

FERGUSON, R. (2012). “Learning analytics : drivers, developments and challenges”. In : *International Journal of Technology Enhanced Learning* 4.5/6, p. 304–317 (cf. p. 12).

FERRARIS, C., A. LEJEUNE, L. VIGNOLLET et J.-P. DAVID (2005). “Modélisation de scénarios d’apprentissage collaboratifs pour la classe”. In : *Environnements Informatiques pour l’Apprentissage Humain*. Montpellier, France, p. 285–296 (cf. p. 26).

FERRÉ, S. (2012). “SQUALL : A Controlled Natural Language for Querying and Updating RDF Graphs”. In : *International Workshop on Controlled Natural Language*. Springer. Berlin, Heidelberg : Springer Berlin Heidelberg, p. 11–25 (cf. p. 195).

FREIRE, J., D. KOOP, E. SANTOS et C. T. SILVA (2008). “Provenance for Computational Tasks : A Survey”. In : *Computing in Science Engineering* 10.3, p. 11–21 (cf. p. 45).

FREIRE, J., C. T. SILVA, S. P. CALLAHAN et al. (2006). “Managing Rapidly-Evolving Scientific Workflows”. In : *International Provenance and Annotation Workshop*. Berlin, Heidelberg : Springer Berlin Heidelberg, p. 10–18 (cf. p. 115, 210).

GAJIRO, D. et Y. GIL (2010). *Open Provenance Model for Workflows Ontology*. URL : http://www.opmw.org/model/OPMW/ (visité le 27 sept. 2018) (cf. p. 47, 48, 50, 110, 114).

GALAXY PROJECT (2018). *Galaxy Project Stat*. URL : https://galaxyproject.org/galaxy-project/statistics/ (visité le 16 août 2018) (cf. p. 39).

GARIJO, D. et Y. GIL (2011). “A new approach for publishing workflows : abstractions, standards, and linked data”. In : *Proceedings of the 6th workshop on Workflows in support of large-scale science*. ACM, p. 47–56 (cf. p. 46, 47, 53).

– (2012). “Augmenting PROV with Plans in P-PLAN : Scientific Processes as Linked Data”. In : *Proceedings of the 2nd International Workshop on Linked Science*. CEUR Workshop Proceedings (cf. p. 45–47, 51–53, 114).

GARTNER (2018). *Planning Guide for Data Analytics*. URL : https://www.gartner.com/binaries/content/assets/events/keywords/catalyst/catus8/2017_planning_guide_for_data_analytics.pdf (visité le 5 oct. 2018) (cf. p. 82).

GIL, Y., E. DEELMAN, M. ELLISMAN et al. (2007). “Examining the Challenges of Scientific Workflows”. In : *Computer* 40.12, p. 24–32 (cf. p. 37).

GOBLE, C. A., J. BHAGAT, S. ALEKSEJEVS et al. (2010). “myExperiment : a repository and social network for the sharing of bioinformatics workflows”. In : *Nucleic acids research* 38.2, p. 677–682 (cf. p. 38, 130, 207).

GOBLE, C. A. et D. C. DE ROURE (2007). “myExperiment : social networking for workflow-using e-scientists”. In : *Proceedings of the 2nd workshop on Workflows in support of large-scale science*, p. 1–2 (cf. p. 38, 48, 55, 84).

GOECKS, J., A. NEKRUTENKO et J. TAYLOR (2010). “Galaxy : a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences”. In : *Genome biology* 11.8, R86 (cf. p. 38, 39, 87, 130).

GOOGLE (2019). *Polymer Project*. URL : https://www.polymer-project.org/ (visité le 16 jan. 2019) (cf. p. 154).

GRELLER, W. et H. DRACHSLER (2012). “Translating learning into numbers : A generic framework for learning analytics”. In : *Journal of Educational Technology & Society* 15.3, p. 42–57 (cf. p. 2, 80).

GROSSMAN, R., S. BAILEY, A. RAMU et al. (1999). “The management and mining of multiple predictive models using the predictive modeling markup language”. In : *Information and Software Technology* 41, p. 589–595 (cf. p. 34).

GRUBER, T. R. (1993). “A translation approach to portable ontology specifications”. In : *Knowledge acquisition* 5.2, p. 199–220 (cf. p. 42).

GUARINO, N. (1998). *Formal ontology in information systems : Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy*. 1st. IOS press (cf. p. 42).

GUAZZELLI, A., K. STATHATOS et M. ZELLER (2009a). “Efficient deployment of predictive analytics through open standards and cloud computing”. In : *ACM SIGKDD Explorations Newsletter* 11.1, p. 32–38 (cf. p. 35).

GUAZZELLI, A., M. ZELLER, W.-C. LIN, G. WILLIAMS et al. (2009b). “PMML : An open standard for sharing models”. In : *The R Journal* 1.1, p. 60–65 (cf. p. 34, 35).

GUIDES IN METROLOGY, J. C. for (2008). *International Vocabulary of Metrology-Basic and General Concepts and Associated Terms*. Rapp. tech. (cf. p. 86).

HÁJEK, P. (1998). *Metamathematics of fuzzy logic*. T. 4. Kluwer (cf. p. 62).

HALLER, A., E. OREN et P. KOTINURMI (2006). “m3po : An Ontology to Relate Choreographies to Workflow Models”. In : *2006 IEEE International Conference on Services Computing (SCC'06)*, p. 19–27 (cf. p. 49).

HAUSENBLAS, M., W. HALB, Y. RAIMOND, L. FEIGENBAUM et D. AYERS (2009). “Scovo : Using statistics on the web of data”. In : *The Semantic Web : Research and Applications*. Springer Berlin Heidelberg, p. 708–722 (cf. p. 51).

HETTNE, K. M., K. WOLSTENCROFT, K. BELHAJJAME et al. (2012). “Best Practices for Workflow Design : How to Prevent Workflow Decay.” In : *SWAT4LS* (cf. p. 48, 50).

HOFMANN, M. et R. KLINKENBERG (2013). *RapidMiner : Data mining use cases and business analytics applications*. Chapman et Hall/CRC (cf. p. 18, 19, 56).

HUFF, D. et I. GEIS (1993). *How to Lie With Statistics*. W. W. Norton & Company (cf. p. 17).

HUMMEL, H. G. K., J. MANDERVELD, C. TATTERSALL et R. KOPER (2004). “Educational modelling language and learning design : new opportunities for instructional reusability and personalised learning”. In : *IJLT* 1, p. 111–126 (cf. p. 26).

HUSSERL, E. (1950). “Idées directrices pour une phénoménologie et une philosophie phénoménologique pures”. In : *Éditions Gallimard* (cf. p. 89).

IBM (2018). *IBM SPSS - User Guide - Statistics Core System*. URL : ftp://public.dhe.ibm.com/software/analytics/spss/documentation/statistics/25.0/en/client/Manuals/IBM_SPSS_Statistics_Core_System_User_Guide.pdf (visité le 13 sept. 2018) (cf. p. 56).

IEEE (2002). *IEEE Standard for Learning Object Metadata - IEEE Std 1484.12.1-2002*. URL : https://standards.ieee.org/findstds/standard/1484.12.1-2002.html (visité le 21 juin 2018) (cf. p. 26).

– (2018). *eScience International Conference*. URL : https://escience-conference.org/ (visité le 15 août 2018) (cf. p. 37).

IHANTOLA, P., A. VIHAVAINEN, A. AHADI et al. (2015). “Educational Data Mining and Learning Analytics in Programming : Literature Review and Case Studies”. In : *Proceedings of the 2015 ITiCSE on Working Group Reports*. ITICSE-WGR ’15. Vilnius, Lithuania : ACM, p. 41–63 (cf. p. 87).

IKSAL, S. (2012). “Observation engineering based on prescription in TEL Environments.” Habilitation à diriger des recherches. Université du Maine (cf. p. 22).

IMS GLOBAL LEARNING CONSORTIUM (2003). *Learning Design specification*. URL : http://www.imsglobal.org/learningdesign/index.html (visité le 21 juin 2018) (cf. p. 26).

– (2018[a]). *Caliper ontology*. URL : https://github.com/IMSGlobal/caliper-ontology (visité le 20 août 2018) (cf. p. 42).

– (2018[b]). *IMS Caliper Analytics Implementation Guide*. URL : https://www.imsglobal.org/caliper/caliperv1p0/ims-caliper-analytics-implementation-guide (visité le 6 août 2018) (cf. p. 27).

– (2018[c]). *IMS Learning Design Best Practice and Implementation Guide*. URL : https://www.imsglobal.org/learningdesign/ldv1p0/imsld_bestv1p0.html (visité le 6 août 2018) (cf. p. 26).

– (2018[d]). *Initial xAPI/Caliper Comparison*. URL : https://www.imsglobal.org/initial-xapicaliper-comparison (visité le 1er juil. 2018) (cf. p. 27).

– (2018[e]). *Real-time, Cross Application Educational Data and Analytics*. URL : https://www.imsglobal.org/initiative/real-time-cross-application-educational-data-and-analytics (visité le 25 juin 2018) (cf. p. 21, 27, 42).

ISO (2005). *ISO/TS 17369 :2005 Statistical data and metadata exchange (SDMX)*. URL : https://www.iso.org/standard/40555.html (visité le 4 sept. 2018) (cf. p. 51).

JEONG, H. et G. BISWAS (2008). “Mining Student Behavior Models in Learning-byTeaching Environments”. In : *Educational Data Mining*, p. 127–136 (cf. p. 17).

KAMBATLA, K., G. KOLLIAS, V. KUMAR et A. GRAMA (2014). “Trends in big data analytics”. In : *Journal of Parallel and Distributed Computing* 74.7, p. 2561–2573 (cf. p. 87).

KÄMPGEN, B. et A. HARTH (2011). “Transforming statistical linked data for use in OLAP systems”. In : *Proceedings of the 7th international conference on Semantic systems*. ACM, p. 33–40 (cf. p. 57).

KERRE, E. E. et M. DE COCK (1999). “Linguistic modifiers : an overview”. In : *Fuzzy logic and soft computing*. Springer, p. 69–85 (cf. p. 62, 138).

KLOOS, C. D., A. ESSA, H. DRACHSLER et P. J. MUÑOZ-MERINO (2018). *Workshop on Applied and Practical Learning Analytics*. URL : http://educate.gast.it.uc3m.es/wapla/ (visité le 8 oct. 2018) (cf. p. 83, 124).

KLUYVER, T., B. RAGAN-KELLEY, F. PÉREZ et al. (2016). “Jupyter Notebooks – a publishing format for reproducible computational workflows”. In : *Positioning and Power in Academic Publishing : Players, Agents and Agendas*. Sous la dir. de F. LOIZIDES et B. SCHMIDT. IOS Press, p. 87–90 (cf. p. 39).

KOEDINGER, K. R., R. S. J. D. BAKER, K. CUNNINGHAM et al. (2010). “A data repository for the EDM community : The PSLC DataShop”. In : *Handbook of educational data mining* 43, p. 43–56 (cf. p. 20, 27, 57).

KOLODNER, J. L. (1992). “An introduction to case-based reasoning”. In : *Artificial Intelligence Review* 6.1, p. 3–34 (cf. p. 211).

KONG WIN CHANG, B., M. LEFEVRE, N. GUIN et P.-A. CHAMPIN (2015). "SPARE-LNC : un langage naturel contrôlé pour l'interrogation de traces d'interactions stockées dans une base RDF". In : *IC2015*. AFIA. Rennes, France (cf. p. 195).

LAFLAQUIÈRE, J. (2009). "Digital trace-based system design in virtual documentary spaces". Thèse de doct. Université de Technologie de Troyes (cf. p. 1, 22).

LAGOZE, C., H. VAN DE SOMPEL, P. JOHNSTON et al. (2008). *Open Archives Initiative Object Reuse and Exchange - Specification*. URL : http://www.openarchives.org/ore/1.0/vocabulary (visité le 6 sept. 2018) (cf. p. 54).

LEARNSPHERE (2018a). *LAK 2018 Workshops*. URL : http://learnsphere.org/workshops.html (cf. p. 29).

– (2018[b]). *A community data infrastructure to support learning improvement online*. URL : http://learnsphere.org/ (visité le 25 juin 2018) (cf. p. 21).

– (2018[c]). *Tigris GitHub*. URL : https://github.com/LearnSphere/WorkflowComponents/tree/master (visité le 7 août 2018) (cf. p. 29).

– (2018[d]). *Tigris : online worflow authoring tool*. URL : https://pslcdatashop.web.cmu.edu/LearnSphere (visité le 3 août 2018) (cf. p. 21, 88, 131, 207).

– (2018[e]). *Tigris Worflow Components*. URL : https://github.com/LearnSphere/WorkflowComponents/blob/master/Workflow%20Components.docx (visité le 7 août 2018) (cf. p. 29).

LEBIS, A. (2016). "Vers une capitalisation des processus d'analyse de traces". In : *Rencontres Jeunes Chercheurs en EIAH (RJC-EIAH 2016)*. Montpellier, France (cf. p. 93, 147, 173).

– (2018a). "Assistance à la réutilisation de processus d'analyse de traces d'apprentissage via une approche narrative et sémantique". In : *Septièmes Rencontres Jeunes Chercheurs en EIAH (RJC EIAH 2018)*. Besançon, France (cf. p. 129, 163).

– (2018[b]). *Analysis Processes Editor - Home Page*. URL : https://perso.liris.cnrs.fr/alexis.lebis/research/iogap/iogap.html (visité le 22 déc. 2018) (cf. p. 147).

– (2018[c]). *Analysis Processes Editor - Prototype*. URL : http://elearning-dev.univ-lyon1.fr/IOGAP/create_gap.html? (visité le 22 déc. 2018) (cf. p. 147).

– (2018[d]). *ATOM - Expérimentation : Données expérimentales*. URL : https://perso.liris.cnrs.fr/alexis.lebis/research/CAPTEN/capten_xp.html (visité le 30 déc. 2018) (cf. p. 181).

– (2018[e]). *CAPTEN - GitHub Home Page*. URL : https://github.com/alexislebis/CAPTEN (visité le 26 déc. 2018) (cf. p. 154, 163, 287).

– (2018[f]). *MANTA - Expérimentation : Données expérimentales*. URL : https://perso.liris.cnrs.fr/alexis.lebis/research/iogap/iogap_xp.html (visité le 30 déc. 2018) (cf. p. 174).

LEBIS, A., M. LEFEVRE, V. LUENGO et N. GUIN (2016). "Towards a Capitalization of Processes Analyzing Learning Interaction Traces". In : *11th European Conference on Technology Enhanced Learning (EC-TEL 2016)*. T. 9891. Lecture Notes in Computer Science. Lyon, France : Springer, p. 397–403 (cf. p. 93, 147, 173).

– (2017a). "Approche narrative des processus d'analyses de traces d'apprentissage : un framework ontologique pour la capitalisation". In : *Environnements Informatiques pour l'Apprentissage Humain*. EIAH 2017. Strasbourg, France (cf. p. 79, 107, 153, 179).

– (2017b). "Capitaliser les processus d'analyse de traces d'e-learning". In : *Méthodologies et outils pour le recueil, l'analyse et la visualisation des traces d'interaction - ORPHEE-RDV*. Font-Romeu, France (cf. p. 93, 147).

– (2018a). "Capitalisation of Analysis Processes : Enabling Reproducibility, Openess and Adaptability thanks to Narration". In : *LAK '18 - 8th International Conference on Learning Analytics and Knowledge*. Sydney, Australia : ACM, p. 245–254 (cf. p. 79, 107, 153, 179).

– (2018[b]). *CAPTEN : Ontology*. URL : https://perso.liris.cnrs.fr/alexis.lebis/research/CAPTEN/ontology.html (visité le 5 nov. 2018) (cf. p. 114, 115).

LEBO, T., S. SAHOO, D. MCGUINNESS et al. (2013). *PROV-O : The PROV Ontology*. URL : https://www.w3.org/TR/prov-o/ (visité le 22 août 2018) (cf. p. 43, 44, 50).

LEE, C.-C. (1990). “Fuzzy logic in control systems : fuzzy logic controller. I”. In : *IEEE Transactions on systems, man, and cybernetics* 20.2, p. 404–418 (cf. p. 62).

LIAW, S.-S. (2008). “Investigating students’ perceived satisfaction, behavioral intention, and effectiveness of e-learning : A case study of the Blackboard system”. In : *Computers & Education* 51.2, p. 864–873 (cf. p. 12).

LIM, C., S. LU, A. CHEBOTKO et F. FOTOUHI (2010). “Prospective and retrospective provenance collection in scientific workflow environments”. In : *2010 IEEE International Conference on Services Computing*. IEEE, p. 449–456 (cf. p. 45).

LUDÄSCHER, B., I. ALTINTAS, C. BERKLEY et al. (2006). “Scientific workflow management and the Kepler system”. In : *Concurrency and Computation : Practice and Experience* 18.10, p. 1039–1065 (cf. p. 50, 51).

LUKAROV, V., M. A. CHATTI, H. THÜS et al. (2014). “Data Models in Learning Analytics.” In : *CEUR Workshop Proceedings* 1227, p. 88–95 (cf. p. 13).

LUND, K. et A. MILLE (2009). “Traces, traces d’interactions, traces d’apprentissages : définitions, modèles informatiques, structurations, traitements et usages”. In : *Analyse de traces et personnalisation des environnements informatiques pour l’apprentissage humain. Hermès*, p. 21–66 (cf. p. 12).

LYKOURENTZOU, I., I. GIANNOUKOS, G. MPARDIS, V. NIKOLOPOULOS et V. LOUMOS (2009). “Early and dynamic student achievement prediction in e-learning courses using neural networks”. In : *Journal of the Association for Information Science and Technology* 60.2, p. 372–380 (cf. p. 15).

MANDRAN, N., M. ORTEGA, V. LUENGO et D. BOUHINEAU (2013). *UnderTracks*. URL : http://projet-undertracks.imag.fr/ (visité le 29 juin 2018) (cf. p. 28, 56, 57).

– (2015). “DOP8 : merging both data and analysis operators life cycles for technology enhanced learning”. In : *Proceedings of LAK’15*. ACM, p. 213–217 (cf. p. 2, 15, 16, 18–21, 27, 28, 35, 38, 42, 84, 85, 160).

MATES, P., E. SANTOS, J. FREIRE et C. T. SILVA (2011). “Crowdlabs : Social analysis and visualization for the sciences”. In : *Scientific and Statistical Database Management*. Springer Berlin Heidelberg, p. 555–564 (cf. p. 38).

MERCERON, A. et K. YACEF (2008). “Interestingness measures for association rules in educational data”. In : *1st International Conference on Educational Data Mining*, p. 57–66 (cf. p. 13).

MICHAELIDES, D. T., R. PARKER, C. CHARLTON, W. J. BROWNE et L. MOREAU (2016). “Intermediate Notation for Provenance and Workflow Reproducibility”. In : *Provenance and Annotation of Data and Processes*. Springer International Publishing, p. 83–94 (cf. p. 46).

MISSIER, P., S. C. DEY, K. BELHAJJAME, V. CUEVAS-VICENTTÍN et B. LUDÄSCHER (2013). “D-PROV : Extending the PROV Provenance Model with Workflow Structure.” In : *5th USENIX Workshop on the Theory and Practice of Provenance (TaPP 13)*. USENIX Association (cf. p. 45–47, 49, 50, 53, 57).

MITX et HARVARDX (2014). *HarvardX-MITx Person-Course Academic Year 2013 De-Identified dataset, version 2.0* (cf. p. 13, 14, 83, 154, 181).

MOREAU, L. et al. (2010). “The foundations for provenance on the web”. In : *Foundations and Trends in Web Science* 2.2–3, p. 99–241 (cf. p. 50).

MOREAU, L., B. CLIFFORD, J. FREIRE et al. (2011). “The open provenance model core specification (v1. 1)”. In : *Future Generation Computer Systems* 27.6, p. 743–756 (cf. p. 48).

MOREAU, L., J. FREIRE, J. FUTRELLE et al. (2008). “The open provenance model : An overview”. In : *International Provenance and Annotation Workshop*. Springer, p. 323–326 (cf. p. 48).

MOREAU, L., P. MISSIER, K. BELHAJJAME et al. (2013). *Prov-dm : The prov data model*. URL : https://www.w3.org/TR/2013/REC-prov-dm-20130430/ (visité le 24 août 2018) (cf. p. 44).

MYEXPERIMENT (2015a). *MyExperiment*. URL : https://www.myexperiment.org (visité le 15 août 2018) (cf. p. 38).

– (2015b). *MyExperiment - About*. URL : https://www.myexperiment.org/about (visité le 16 août 2018) (cf. p. 39).

MYGRID (2008). *myGrid - About us*. URL : http://www.mygrid.org.uk/about-us/ (visité le 15 août 2018) (cf. p. 38).

NETWORK WORKING GROUP (2005). *Common Format and MIME Type for Comma-Separated Values (CSV) Files*. URL : https://tools.ietf.org/html/rfc4180#page-2 (visité le 21 juin 2018) (cf. p. 12).

NKAMBOU, R., R. MIZOGUCHI et J. BOURDEAU (2010). *Advances in intelligent tutoring systems*. T. 308. Springer Science & Business Media (cf. p. 12).

NORUŠIS, M. J. (1990). *SPSS base system user's guide*. Prentice Hall (cf. p. 18, 28).

NUNN, S., J. AVELLA, T. KANAI et M. KEBRITCHI (2016). "Learning Analytics Methods, Benefits, and Challenges in Higher Education : A Systematic Literature Review." In : *Online Learning* 20.2, p. 13–29 (cf. p. 15, 17).

OBERLE, D., S. LAMPARTER, S. GRIMM et al. (2006). "Towards ontologies for formalizing modularization and communication in large software systems". In : *Applied Ontology* 1.2, p. 163–202 (cf. p. 46).

OBJECT MANAGMENT GROUP (2011). *Business Process Model and Notation Specification Version 2.0*. URL : https://www.omg.org/spec/BPMN/2.0 (visité le 16 août 2018) (cf. p. 39).

OINN, T., M. ADDIS, J. FERRIS et al. (2004). "Taverna : a tool for the composition and enactment of bioinformatics workflows". In : *Bioinformatics* 20.17, p. 3045–3054 (cf. p. 38, 48).

OINN, T., M. GREENWOOD, M. ADDIS et al. (2006). "Taverna : lessons in creating a workflow environment for the life sciences". In : *Concurrency and Computation : Practice and Experience* 18.10, p. 1067–1100 (cf. p. 38, 39, 41).

O'REILLY, U.-M. et N. HOFFMAN (2017). *Sharing In LearnSphere, Study Report*. Rapp. tech. (cf. p. 29).

PAGE, K., R. PALMA, P. HOLUBOWICZ et al. (2012). "From workflows to Research Objects : an architecture for preserving the semantics of science". In : *Proceedings of the Second International Workshop on Linked Science*. T. 10. Citeseer (cf. p. 48, 108, 153).

PAPAMITSIOU, Z. et A. A. ECONOMIDES (2014). "Learning Analytics and Educational Data Mining in Practice : A Systematic Literature Review of Empirical Evidence". In : *Journal of Educational Technology & Society* 17.4, p. 49–64 (cf. p. 14, 15, 17).

PEREZ, F. et B. E. GRANGER (2007). "IPython : A System for Interactive Scientific Computing". In : *Computing in Science Engineering* 9.3, p. 21–29 (cf. p. 39).

PERNIN, J.-P. (2004). "LOM, SCORM et IMS-Learning Design : ressources, activités et scénarios". In : *Colloque sur L'indexation des ressources pédagogiques numériques*. T. 16 (cf. p. 26).

PIVARSKI, J., C. BENNETT et R. L. GROSSMAN (2016). "Deploying analytics with the portable format for analytics (PFA)". In : *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, p. 579–588 (cf. p. 36, 37).

PRAXADEMIA (2015). *Formaliser la connaissance*. URL : http://www.praxademia.com/wp-content/uploads/2015/03/FormaliserLaConnaissance_FR.pdf (visité le 22 août 2018) (cf. p. 42).

PROJET HUBBLE (2016). *HUman oBservatory Based on anaLysis of E-learning traces (HUBBLE)*. URL : http://hubblelearn.imag.fr/?lang=fr (visité le 30 sept. 2018) (cf. p. 29, 81, 88, 91, 131, 160).

– (2018). *HUBBLE Plateforme*. URL : https://hubble-lium.univ-lemans.fr/ (visité le 13 mar. 2019) (cf. p. 29, 91, 114, 207).

QUEIRÓS, R. et J. P. LEAL (2013). "A Survey on eLearning Content Standardization". In : *Information Systems, E-learning, and Knowledge Management Research*. Springer Berlin Heidelberg, p. 433–438 (cf. p. 26).

R DEVELOPMENT CORE TEAM (2008). *R : A Language and Environment for Statistical Computing*. ISBN 3-900051-07-0. R Foundation for Statistical Computing. Vienna, Austria (cf. p. 15, 19, 85, 160).

RAPIDMINER (2018[a]). *Global Search in RapidMiner Studio*. URL : https://docs.rapidminer.com/latest/studio/global-search (visité le 29 juin 2018) (cf. p. 56).

– (2018[b]). *RapidMiner Studio Manual*. URL : https://docs.rapidminer.com/downloads/RapidMiner-v6-user-manual.pdf (visité le 29 juin 2018) (cf. p. 19, 56).

REFFAY, C., M.-L. BETBEDER et T. CHANIER (2012). "Multimodal learning and teaching corpora exchange : lessons learned in five years by the Mulce project". In : *International Journal of Technology Enhanced Learning* 4.1-2, p. 11–30 (cf. p. 19, 27).

ROMERO, C., J. R. ROMERO et S. VENTURA (2014). “A Survey on Pre-Processing Educational Data”. In : *Educational Data Mining : Applications and Trends*. Springer International Publishing, p. 29–64 (cf. p. 16).

ROMERO, C. et S. VENTURA (2010). “Educational Data Mining : A Review of the State of the Art”. In : *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 40.6, p. 601–618 (cf. p. 16).

ROMERO, C., S. VENTURA et E. GARCÍA (2008). “Data mining in course management systems : Moodle case study and tutorial”. In : *Computers & Education* 51.1, p. 368–384 (cf. p. 12).

ROMERO, C., S. VENTURA, M. PECHENIZKIY et R. S. J. d. BAKER (2010a). *Handbook of Educational Data Mining*. CRC Press (cf. p. 1, 82).

– (2010b). “Process Mining from Educational Data”. In : *Handbook of Educational Data Mining*. CRC Press, p. 139–158 (cf. p. 19).

ROMERO, C., S. VENTURA, A. ZAFRA et P. DE BRA (2009). “Applying Web usage mining for personalizing hyperlinks in Web-based adaptive educational systems”. In : *Computers & Education* 53.3, p. 828–840 (cf. p. 15).

ROSCH, E. H. (1973). “Natural categories”. In : *Cognitive psychology* 4.3, p. 328–350 (cf. p. 90, 94, 95).

SANCHEZ, E. (2006). *Fuzzy logic and the semantic web*. T. 1. Elsevier Science (cf. p. 59).

SCHEIDEGGER, C., D. KOOP, E. SANTOS et al. (2008). “Tackling the provenance challenge one layer at a time”. In : *Concurrency and Computation : Practice and Experience* 20.5, p. 473–483 (cf. p. 44).

SETTOUTI, L., N. GUIN, V. LUENGO et A. MILLE (2010). “Trace-Based Learner Modelling Framework for Technology-Enhanced Learning Systems”. In : *IEEE International Conference on Advanced Learning Technologies*. Sous la dir. de COMPUTER SOCIETY PUBLICATIONS, IEEE. Sousse, Tunisia (cf. p. 31).

– (2011). “Adaptable and Reusable Query Patterns for Trace-Based Learner Modelling”. In : *EC-TEL 2011 - 6th European Conference on Technology Enhanced Learning : towards ubiquitous learning*. Palermo, Italy : Springer, p. 384–397 (cf. p. 31).

SETTOUTI, L. S., Y. PRIÉ, A. MILLE et J.-C. MARTY (2006). “Systèmes à base de traces pour l’apprentissage humain”. In : *Communication in the international TICE, Technologies de l’Information et de la Communication dans l’Enseignement Supérieur et l’Entreprise* (cf. p. 12, 22, 23).

SIEMENS, G. (2005). *Connectivism : A learning theory for the digital age*. URL : http://www.itdl.org/journal/jan_05/article01.htm (visité le 18 sept. 2018) (cf. p. 59).

– (2012). “Learning analytics : envisioning a research discipline and a domain of practice”. In : *Proceedings of the 2nd international conference on learning analytics and knowledge*. ACM, p. 4–8 (cf. p. 12).

SIEMENS, G. et R. S. J. D. d BAKER (2012). “Learning analytics and educational data mining : towards communication and collaboration”. In : *Proceedings of the 2nd international conference on learning analytics and knowledge*. ACM, p. 252–254 (cf. p. 2).

SIEMENS, G., D. GASEVIC, C. HAYTHORNTHWAITE et al. (2011). *Open Learning Analytics : an integrated & modularized platform*. Rapp. tech. SolAR (cf. p. 26, 28).

SIEMENS, G. et P. LONG (2011). “Penetrating the Fog : Analytics in Learning and Education.” In : *EDUCAUSE review* 46.5, p. 30–40 (cf. p. 15).

SLADE, S. et P. PRINSLOO (2013). “Learning analytics : Ethical issues and dilemmas”. In : *American Behavioral Scientist* 57.10, p. 1510–1529 (cf. p. 2).

SMEULDERS, A. W., M. WORRING, S. SANTINI, A. GUPTA et R. JAIN (2000). “Content-based image retrieval at the end of the early years”. In : *IEEE Transactions on Pattern Analysis & Machine Intelligence* 12, p. 1349–1380 (cf. p. 50, 59).

SOLDATOVA, L. N. et R. D. KING (2006). “An ontology of scientific experiments”. In : *Journal of the Royal Society Interface* 3.11, p. 795–803 (cf. p. 51).

SOWA, J. F. (1984). *Conceptual Structures : Information Processing in Mind and Machine*. Boston, MA, USA : Addison-Wesley Longman Publishing Co., Inc. (cf. p. 60).

SPRINGER NATURE (2016). “Reality check on reproducibility”. In : t. 533, p. 437 (cf. p. 18, 86).

STAAB, S. et R. STUDER (2010). *Handbook on ontologies*. Springer Science & Business Media (cf. p. 42, 109).

STAMPER, J., K. KOEDINGER, P. PAVLIK JR et al. (2016). "Educational Data Analysis Using LearnSphere Workshop". In : *9th International Conference on Educational Data Mining - Workshop* (cf. p. 29, 35, 38).

STAMPER, J. C., K. R. KOEDINGER, R. S. J. D. d BAKER et al. (2011). "Managing the Educational Dataset Lifecycle with DataShop". In : *International Conference on Artificial Intelligence in Education*. Springer Berlin Heidelberg, p. 557–559 (cf. p. 18, 20, 31, 33, 80, 97).

STATO (2018[a]). *Correlation coefficient semantic*. URL : https://www.ebi.ac.uk/ols/ontologies/stato/terms?iri=http%3A%2F%2Fpurl.obolibrary.org%2Fobo%2FSTATO_0000142 (visité le 5 nov. 2018) (cf. p. 116).

– (2018[b]). *The Statistics Ontology*. URL : http://frog.oerc.ox.ac.uk:8080/stato-app/ (visité le 30 août 2018) (cf. p. 49).

SUTHERS, D. et D. ROSEN (2011). "A Unified Framework for Multi-level Analysis of Distributed Learning". In : *Proceedings of the 1st International Conference on Learning Analytics and Knowledge*. LAK '11. Banff, Alberta, Canada : ACM, p. 64–74 (cf. p. 2).

TERDJIMI, M. (2019). *HyLAR-Reasoner v.1.8.3*. URL : https://github.com/ucbl/HyLAR-Reasoner/tree/f7a6b1ddbde1f42f209a45a7e68cdbfea793d9cd (visité le 7 fév. 2019) (cf. p. 197).

TERDJIMI, M., L. MÉDINI et M. MRISSA (2015). "HyLAR : Hybrid Location-Agnostic Reasoning". In : *ESWC Developers Workshop 2015*. Portoroz, Slovenia, p. 1 (cf. p. 163).

TRILLAS, E. et L. ECIOLAZA (2015). *Fuzzy logic : an introductory course for engineering students*. T. 320. Springer (cf. p. 61).

TSANTIS, L. et J. CASTELLANI (2001). "Enhancing learning environments through solution-based knowledge discovery tools : Forecasting for self-perpetuating systemic reform". In : *Journal of Special Education Technology* 16.4, p. 39–52 (cf. p. 80).

VAHDAT, M. (2017). "Learning analytics and educational data mining for inquiry-based learning". English. Proefschrift. Thèse de doct. Department of Industrial Design, University of Geneva (cf. p. 31).

VAHDAT, M., A. GHIO, L. ONETO et al. (2015). "Advances in learning analytics and educational data mining". In : *ESANN 2015 proceedings : European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*. Katholieke Universiteit Leuven, p. 297–306 (cf. p. 12).

VAN HARMELEN, F., V. LIFSCHITZ et B. PORTER (2008). *Handbook of knowledge representation*. T. 1. Elsevier (cf. p. 55).

VERBERT, K., H. DRACHSLER, N. MANOUSELIS et al. (2011). "Dataset-driven research for improving recommender systems for learning". In : *Proceedings of the 1st International Conference on Learning Analytics and Knowledge*. ACM, p. 44–53 (cf. p. 19).

VERBERT, K., N. MANOUSELIS, H. DRACHSLER et E. DUVAL (2012). "Dataset-Driven Research to Support Learning and Knowledge Analytics". In : *Journal of Educational Technology & Society* 15.3, p. 133–148 (cf. p. 15, 19, 31, 32).

VIDAL, J. C., T. RABELO et M. LAMA (2015). "Semantic Description of the Experience API Specification". In : *2015 IEEE 15th International Conference on Advanced Learning Technologies*, p. 268–269 (cf. p. 43).

VOLLE, M. (2001). *Le métier de statisticien*. URL : http://www.volle.com/index.html?id=0 (visité le 20 juin 2018) (cf. p. 16, 57, 80, 82).

W. BRUCE CROFT Donald Metzler, T. S. (2009). *Search Engines : Information Retrieval in Practice*. Pearson (cf. p. 57, 59).

W3C (2012). *OWL 2 Web Ontology Language Document Overview (Second Edition)*. URL : https://www.w3.org/TR/2012/REC-owl2-overview-20121211/ (visité le 20 août 2018) (cf. p. 42).

– (2014a). *RDF Schema 1.1*. URL : https://www.w3.org/TR/rdf-schema/ (visité le 20 août 2018) (cf. p. 42).

– (2014b). *RDF Schema - Syntax NS*. URL : http://www.w3.org/1999/02/22-rdf-syntax-ns (visité le 7 déc. 2018) (cf. p. 117).

– (2018). *World Wide Web Consortium*. URL : https://www.w3.org/ (visité le 24 août 2018) (cf. p. 44).

WAGNER, E. et P. ICE (2012). "Data changes everything : Delivering on the promise of learning analytics in higher education." In : *EDUCAUSE Review* 47.4, p. 32–42 (cf. p. 26).

WHITE, T. (2009). *Hadoop : The Definitive Guide*. 1st. O'Reilly Media, Inc. (cf. p. 36).

WIDYANTORO, D. H. et J. YEN (2001). "A fuzzy ontology-based abstract search engine and its user studies". In : *IEEE International Conference on Fuzzy Systems*. T. 3. IEEE, p. 1291–1294 (cf. p. 62).

WITTEN, I. H., E. FRANK, M. A. HALL et C. J. PAL (2016). *Data Mining : Practical machine learning tools and techniques*. Morgan Kaufmann (cf. p. 18, 19, 28, 160).

WROE, C., C. GOBLE, A. GODERIS et al. (2007). "Recycling workflows and services through discovery and reuse". In : *Concurrency and Computation : Practice and Experience* 19.2, p. 181–194 (cf. p. 38).

YEN, J. (1991). "Generalizing Term Subsumption Languages to Fuzzy Logic". In : *Proceedings of the 12th International Joint Conference on Artificial Intelligence*. T. 1. IJCAI'91. Sydney, New South Wales, Australia : Morgan Kaufmann Publishers Inc., p. 472–477 (cf. p. 59, 62).

ZADEH, L. A. (1965). "Fuzzy sets". In : *Information and Control* 8.3, p. 338–353 (cf. p. 61, 134).

ZADEH, L. A. (1971). "Quantitative fuzzy semantics". In : *Information sciences* 3.2, p. 159–176 (cf. p. 62, 134, 138).

Partie VII

Annexes

A

Exemple de fiches techniques pour l'élaboration des opérateurs indépendants

A.1 Fiche resumée d'un opérateur implémenté

Fiche resumée d'un opérateur, illustrant notre démarche pour formaliser les caractéristiques d'un opérateur implémenté - ici dans kTBS, en vue de définir nos méta-modèles.

Filtre hybride simple (dans kTBS)

Formalisme

$$F_{hys} = (N, T_M,\ F_{tmp} \setminus \{N, T_M\},\ F_{sts} \setminus \{N, T_M\})$$
$$F_{hys} = (N, T_M, t_b, t_e, \zeta),\ t_b \in \aleph,\ t_e \in \aleph,\ \zeta = (C_1, ..., C_n).$$
$$F_{hys} :\ M_T \rightarrow M_T$$

Objectif

Filtrer les obsels en fonction du ou des type(s) spécifié(s), sur un domaine temporel spécifié.

Description

Généralisation du filtre temporel et du filtre structurel simple, le filtre hybride permet d'effectuer un filtrage sur le type des obsles ainsi que sur leur position temporelle.
En omettant tous les paramètres à ce filtre, il peut aussi bien servir d'opérateur de copie.
Autrement, il peut effectuer un filtre temporel avec l'omission des paramètres concernant la structure, et inversement : il peut effectuer un filtre structurel simple *via* omission des bornes temporelles.
Ce filtre crée une abstraction supplémentaire pour l'utilisateur et lui évite ainsi d'avoir à recourir à l'enchaînement d'un filtre temporel et structurel simple (ou inversement). Il s'agit donc d'une opération composite, qu'il est possible de scinder en deux (i.e. filtre temporel et structurel simple).

Entrée

Au moins la trace source. Si aucune information temporelle ou de type n'est fournie, il s'agit d'une simple copie de la trace.
ζ, les types d'obsels à conserver.
t_b, la valeur qui est comparée avec le champ *begin* de l'obsel, répondant à *after* dans la documentation.
t_e, la valeur qui est comparée avec le champ *end* de l'obsel, répondant à *before* dans la documentation.
Si ζ est nul, il s'agit d'un filtre temporel.
Si t_b et t_e sont nuls, il s'agit d'un filtre structurel.

Sortie

Une nouvelle trace, nommée *N,* contenant tous les obsels qui ont passé le filtre.

Algorithme subodoré

Entrée *: obs* $\leftarrow$ *Obsel,* t_b, t_e, ζ
Sortie *: obs*
obs $\leftarrow$ ***call*** *FiltreTemporel(obs*, t_b, t_e*)*
obs $\leftarrow$ ***call*** *FiltreStructurel(obs*, ζ*)*
retourner *obs*

A.2 Fiche d'identification d'un concept d'opération

Fiche d'identification du concept d'opération illustrative de notre démarche pour formaliser un concept d'opération, en vue de définir nos méta-modèles.

Filtre Conditionnel

Formalisme

$$F_{val} = (\ variable, val, operande)$$

Objectif

Filtrer les éléments contenues dans la variable $variable$ des traces en fonction d'une condition.

Description

Filtre Conditionnel applique un test sur les éléments de la variable considérée, et laisse passer tous ceux qui le respectent. La valeur peut être numérique ou textuelle, et l'opérande est choisie parmi une liste contenant : <, ≤,≥,>,!=,==.

Entrée

$variable$, la variable contenant les éléments à analyser,
val, la valeur à respecter,
$operande$, l'opérande de la condition à choisir parmi <, ≤,≥,>,!=,==.

Sortie

Une nouvelle trace basée sur la trace d'entrée dont est issue $variable$ (*i.e.* les éléments indépendants à cette variable sont conservés). Cette nouvelle trace possède également la variable $variable\ \ categorie_{elem}$, privée des éléments qui n'ont pas passé le filtre.

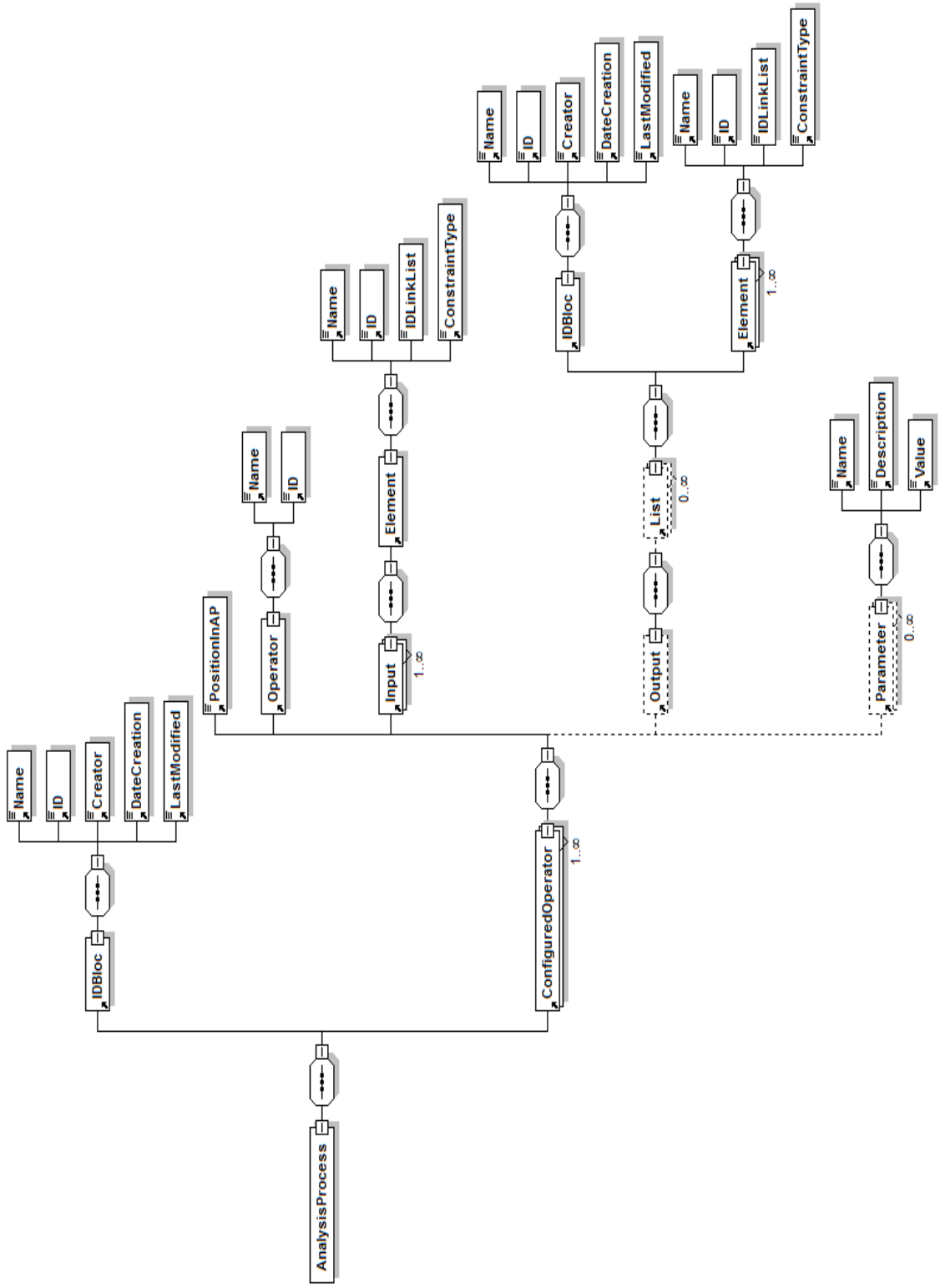
Processus

- $F_{approx\_kTBS} = (N, T_{M\_element}, D_{element}, val, operande)$
- $F_{approx\_UT} = (\tau_{element}, \Psi_{element}, val, operande)$

méta-modèle des processus d'analyse indépendant

B

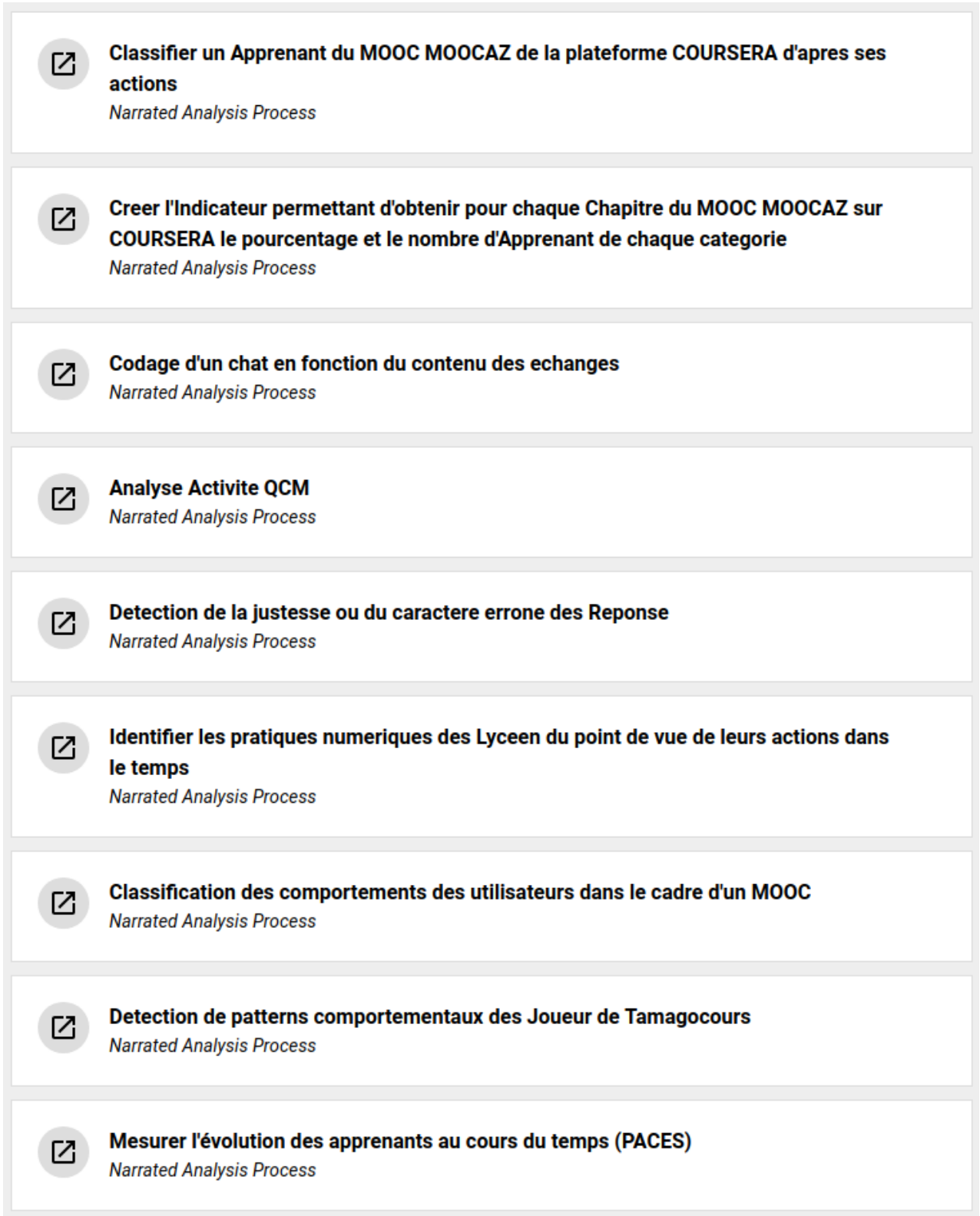
Vue schématique, ici complète, du Document Type Definition (DTD) utiliser pour définir le méta-modèle d'un processus d'analyse indépendant, introduit dans le Chapitre 7.



Liste des processus d'analyse narrés

C

Capture d'écran montrant les neuf analyses utilisées pour peupler le prototype **CAPTEN-TORTOISE**.



Documents liés à l'expérimentation des concepts de CAPTEN-ALLELE

D

D.1 Protocole

Protocole expérimental
07/01/2016

Introduction

L'outil que vous allez utiliser est un prototype expérimental permettant de tester une approche théorique : permettre d'exprimer des processus d'analyse de traces d'apprentissage indépendamment de toute plate-forme d'analyse.

Ce prototype n'étant pas toujours intuitif, la première partie va permettre de vous familiariser avec l'outil. La deuxième partie va vous proposer d'exprimer par vous-même un processus d'analyse spécifique. Dans les deux cas, vous n'aurez qu'à décrire le processus d'analyse, sans procéder à son implémentation dans une une plate-forme d'analyse de traces spécifique.

Il est à noter que l'outil est encore en phase de développement, que des bugs subsistent encore, et que certaines fonctionnalités (comme des opérateurs) ne sont pas encore implémentées.

Étape préalable

Objectif

Avant de commencer à utiliser l'outil, cette étape préalable consiste à exprimer un processus d'analyse sur papier, comme vous le feriez en temps normal pour réfléchir au problème posé par un demandeur.

Cette étape a pour but d'étudier la manière dont les analystes et les statisticiens conceptualisent le besoin d'un demandeur, avant d'entamer la récolte des données et leur manipulation pour arriver au but escompté par ledit demandeur.

Nous vous proposons donc de décrire sur papier un processus d'analyse simple, de votre choix, que nous vous demanderons de reproduire un peu plus tard dans l'outil.

Si vous n'avez pas d'idée, nous vous proposons de travailler sur le processus d'analyse qui porte sur un indicateur : **"Pourcentage, pour la semaine dernière, de visionnage d'une vidéo donnée"**.

La page suivante vous permet de décrire votre processus.

Description du processus

Objectif du processus :

Complexité (subjectif, selon vous) du processus :

Description des étapes du processus (textuelle, graphique ou les deux) :

Se familiariser avec l'outil

Description de l'outil

L'outil que vous allez utiliser tente d'offrir aux analystes une méthode pour exprimer les processus d'analyse sans avoir à se soucier des différentes plates-formes d'analyse à utiliser.
Pour ce faire, vous ne manipulerez pas directement les données aux travers des opérateurs comme vous pouvez en avoir l'habitude, mais plutôt le type de ces données. Par exemple, vous indiquerez qu'il faut un nom, une date, un score… mais jamais "Dupont". De cette manière, vous fixez des pré-requis sur le contenu de vos données.
Pour décrire votre processus, vous utiliserez des opérateurs, qui prennent en entrée certains types de données et qui en produisent d'autres en sortie.

Il est important que vous compreniez que la partie de l'outil que vous allez utiliser est uniquement descriptive et n'a pas vocation à calculer : elle ne sert qu'à exprimer un processus. L'instanciation – ou l'application de la description du processus à un vrai cas d'étude – pour indiquer/effectuer les démarches nécessaires et réaliser le calcul est envisagée dans une autre partie qui sera développée à terme.

Cas 1 : Prise en main (durée 30 min.)

Dans cet exemple, qui va vous aider à comprendre la logique de fonctionnement de l'outil, nous allons essayer de représenter le processus d'analyse de l'**indicateur "nombre de personnes connectées à une plate-forme dans un intervalle de temps précis"**.
Nous allons procéder étape par étape, mais avant cela, il faut réfléchir à comment obtenir notre processus.
Pour l'obtenir, on peut procéder de cette manière : "Il faut d'abord isoler l'intervalle de temps. Puis, compter les apprenants qui ont passé le filtre".

1. Présentation de l'outil

Chargez la page "create_gap.html" présente sur le support USB ou à l'adresse http://elearning-dev.univ-lyon1.fr/IOGAP/create_gap.html?. Vous devez avoir accès à internet pour que l'outil fonctionne.

C'est dans cette interface que vous évoluerez. Considérez-la comme le "cahier de brouillon" dont vous vous servirez pour exprimer votre réflexion. La 1ère zone grise, en dessous du titre "Créateur de Processus d'Analyse Générique", représente les ressources à votre disposition (listes initiales, opérateurs, listes calculées). La 2nde zone grise représente votre processus d'analyse en cours de construction (partie gauche), ainsi que sa représentation graphique (partie droite).
La dernière zone, cachée par le bouton "Sélection des étapes", permet de sélectionner les étapes qui constitueront au final le processus d'analyse.

2. Initialisation du processus

Dans la 2nde zone, cliquez sur l'onglet "**+**" afin d'indiquer au système que vous souhaitez commencer à créer un processus d'analyse. Vous verrez apparaître un nouvel onglet "Feuille 1" contenant 3 colonnes (Inputs, Operator, Outputs).

3. Définition des données relatives aux traces d'apprentissage

La première chose à faire consiste à définir les *données nécessaires* (ou plutôt le type des données) que vous utiliserez. Cliquez sur le bouton "**Listes Initiales**". Vous verrez apparaître une nouvelle fenêtre.

Les listes initiales représentent les données nécessaires dont vous avez besoin pour effectuer votre processus d'analyse. Elles peuvent être utilisées à n'importe quelle étape du processus.

Cliquez sur le bouton "**Créer une nouvelle liste**", et remplissez les champs requis avec les informations relatives à vos types de données. Dans notre exemple, nous pouvons ainsi créer une liste "MOOC_avec_video". N'oubliez pas de valider. La liste apparaît alors avec 0 élément.

Une donnée nécessaire est toujours contenue dans une liste. Cette étape de création d'au moins une liste est obligatoire. Une liste représente ce que doit impérativement contenir votre trace.

Cliquez sur la liste qui vient d'apparaître. Vous pouvez maintenant ajouter des éléments à cette liste. Il faut en ajouter deux (le temps et les apprenants). Cliquez sur "**Ajouter un élément**" pour ajouter le premier élément. Recommencez l'opération pour le deuxième.

En cliquant deux fois sur "Ajouter un élément", vous spécifiez donc deux données nécessaires impératives à votre processus d'analyse. Notez que lors de leur création, les nouvelles données ont un nom attribué par défaut du style "autoElementX".

Pour sélectionner une donnée nécessaire, il suffit de cliquer dessus.

Le fait de sélectionner un élément permet d'appliquer les actions "Supprimer l'élément", "Utiliser l'élément" et "Modifier l'élément".

Modifiez le nom des deux données créées. Appelez par exemple l'une "temps", l'autre "idApprenant".

La zone de modification ne s'ouvre qu'après sélection d'un élément. Notez que renommer

convenablement vos éléments permet de mieux vous organiser. C'est la sémantique que donne l'utilisateur aux données nécessaires qu'il utilise qui apporte un sens au processus.

L'attribut *type* n'a pas encore d'impact, vous pouvez laisser xs:type.

4. Ajouter un opérateur

La première étape de notre processus d'analyse étant un filtre temporel sur des données temporelles, il faut ajouter "temps" à l'étape courante via "**Utiliser l'élément**" (bouton vert). Vous pouvez fermer la fenêtre.

Si vous ajoutez malencontreusement plus d'un élément, vous pouvez retirer ceux en trop en fermant la fenêtre, en sélectionnant – pour chaque élément indésirable – l'élément à supprimer puis en cliquant sur "Supprimer l'élément sélectionné".

Par ailleurs, quand vous fermez la fenêtre des listes initiales, vos données sont conservées et pourront être réutilisées pour un autre opérateur. Pas d'inquiétude, donc.

Ensuite, il nous faut matérialiser l'action que l'on va appliquer sur ce type de données. Il s'agit ici d'un filtre temporel. Cliquez sur le bouton "**Opérateurs**". Une nouvelle fenêtre s'ouvre.

"Opérateurs" recense toutes les actions (opérations) que notre outil est capable de gérer (actuellement) et qu'il sera capable d'instancier dans les plates-formes connues d'analyse de traces.

Cliquez sur l'opérateur "Filtre temporel" pour le sélectionner.

Lorsque vous sélectionnez un opérateur dans la liste, toutes ses informations sont accessibles, notamment le nombre de données nécessaires qu'il attend en entrée.

Utilisez l'opérateur sélectionné en cliquant sur "**Utiliser cet opérateur**".

Le fait d'avoir suffisamment d'entrées pour un opérateur dans une étape génère automatiquement le résultat.

Le fait de cliquer sur le bouton ajoute directement l'opérateur à l'étape courante. Vous ne pouvez ajouter qu'un seul opérateur à chaque fois.
Si vous désirez remplacer l'opérateur de l'étape courante, il suffira d'utiliser un autre opérateur qui remplacera l'ancien.

Cliquez sur "ok" pour fermer la fenêtre d'alerte, puis cliquez sur le bouton "Fermer". Vous revenez alors sur l'interface principale.

Un opérateur génère toujours une ou plusieurs listes calculées.

Ici, la liste de données calculées contiendra les mêmes éléments que l'ancienne. Ce n'est pas toujours le cas.
Dans notre exemple, nous appliquons un filtre temporel : cette action n'est pas destructrice sur les types de données présents dans les traces, uniquement sur les données qu'ils contiennent.

5. Configurer l'opérateur

Afin de configurer l'opérateur pour cibler la période, cliquez sur le bouton vert de l'opérateur dans l'étape courante. Une nouvelle fenêtre s'ouvre.

En cliquant sur un opérateur dans une étape, vous avez accès à un récapitulatif vous permettant de savoir si oui ou non l'opérateur est bien configuré.
Lorsqu'il manque des informations capitales (ou qu'il est surconfiguré), le bouton "Opérateur" est de couleur orange.

Cliquez sur "**Configuration**".

Il y a deux manières de paramétrer un opérateur. La première est celle que nous allons utiliser. La seconde pourra s'effectuer lors de l'instanciation.

Mettons la date de cette semaine, le format étant en JJ/MM/AAAA. Le paramètre 0 sera donc à JJ-7/MM/AAAA et le paramètre 1 à JJ/MM/AAAA. N'oubliez pas de valider les changements.

Dans la prochaine version du prototype, les paramètres et leurs noms seront plus explicites.

Les paramètres vont être utilisés lors de l'instanciation. Ils n'ont pas d'impact dans la zone de création du processus.

Remarquez qu'il est possible de changer le nom de la liste produite en double-cliquant sur la liste dans la colonne Outputs.

Vous venez de finir la première étape de votre processus d'analyse.

6. Ajouter un autre opérateur

Maintenant, place à la deuxième et dernière étape. Cliquez sur "**+**", à côté de "Feuille 1".

> Vous pouvez créer autant d'étapes que vous le voulez. Celles incomplètes sont indiquées en orange. Notez que vous pouvez également les supprimer.
>
> Vous pouvez renommer les étapes *via* le bouton "Renommer l'étape courante".

Le second opérateur de notre processus d'analyse devra compter le nombre d'apprenants restant dans le créneau temporel défini. Pour ce faire, il ne faut pas utiliser "idApprenant" de la liste initiale, mais celui de la liste générée à l'étape 1.
Cliquez sur "**Listes calculées**".

> Si vous aviez utilisé l'"idApprenant" de votre liste initiale, vous auriez compté tous les apprenants, qu'importe la date associée. Là, vous utilisez les "idApprenant" résultant du filtre temporel.

Une fenêtre s'ouvre avec la liste produite à l'étape 1. Un simple clic sur le nom de la liste affiche les informations de cette liste, un double clic fait apparaître les options de manipulation de la liste. Sélectionnez "idApprenant" et utilisez-le dans l'étape 2, puis fermez la fenêtre.

> Les listes calculées vous permettent donc de réutiliser les productions des différents opérateurs, et ainsi les enchaîner pour créer votre description du processus d'analyse.
>
> À l'étape N, vous pouvez utiliser toutes les productions des étapes N-1. Vous pouvez créer de cette manière des processus complexes.
>
> Si vous avez ajouté trop de fois un élément, ou que vous l'avez ajouté dans une mauvaise étape, vous pouvez toujours le supprimer.

Il faut maintenant compter les apprenants. Ouvrez le menu "**Opérateurs**" et utilisez "**Compter Distinct**".

> "Compter Distinct" est un opérateur permettant de ne compter qu'une fois le même élément. Ainsi, lors de l'instanciation, s'il y a 4 fois l'apprenant avec l'id 777, il ne sera compté qu'une seule fois.

7. Aperçu graphique du processus

Lorsque vous configurez une étape, le graphe du processus que vous êtes en train de définir se met à jour pour vous permettre d'avoir un meilleur aperçu du processus. Ce graphe est visible sur la partie droite de l'interface. Vous pouvez interagir sur les noeuds du graphe en les sélectionnnant.

Dans ce graphe, un cercle est la représentation d'une "feuille" (i.e. étape) du processus d'analyse. Toutes les feuilles, entièrement configurées ou non, seront représentées par des cercles portant leur nom. Si une feuille utilise des données issues d'une autre feuille, alors cette dépendance est matérialisée par une flèche entre les deux cercles qu'il faut lire "B utilise au moins une donnée de A". Sur une flèche entre un cercle A et un cercle B apparaît le nom des plates-formes qui supportent l'opérateur défini dans la feuille A. De plus vous avez la possibilité sous le graphique de choisir une "Visualisation spécifique" en fonction des plates-formes d'analyse de traces supportant les opérations. Les étapes non supportées sont grisées et les étapes non configurées ne sont pas affichées.

8. Finalisation du processus d'analyse

Vous avez terminé d'exprimer toutes les étapes de votre processus d'analyse. Maintenant, il faut choisir lesquelles conserver. Pour ce faire, cliquez sur "Sélection des étapes" en bas de la page et choisissez les deux étapes créées puis sauvegardez.

La possibilité de sélection permet de créer des étapes brouillonnes et de ne pas les prendre en compte quand vient le temps d'exporter le processus d'analyse. Le système sélectionne automatiquement les étapes dépendantes d'une étape choisie, et ainsi de suite. Aucun risque d'oublier une étape importante de cette manière !

Félicitations, vous venez de créer votre première description de processus d'analyse de manière indépendante des plates-formes d'analyse de traces.

L'étape d'instanciation de ce processus n'est pas implémentée pour l'instant, mais devrait par la suite vous permettre de décrire l'indicateur produit, de lier ce processus avec vos traces réelles et d'avoir la notice permettant de l'instancier sur la ou les plate(s)-forme(s) que vous aurez choisie(s).

Questionnaire à remplir.

Cas 2 : Mise en oeuvre de votre processus d'analyse

Maintenant, vous allez tenter de mettre en oeuvre par vous-même le processus que vous avez décrit dans la première partie de l'expérimentation.
Le but est de matérialiser, dans l'outil, la réflexion que vous avez eue lorsque vous avez créé votre processus d'analyse.

Si vous avez des difficultés pour formaliser votre processus, vous pouvez vous référer à la section ci-dessous. Elle se concentre sur l'exemple du **"Pourcentage, pour la semaine dernière, de visionnage d'une vidéo donnée"** :

Pour obtenir ce processus, on peut procéder de cette manière : "Il faut d'abord isoler la semaine désirée. Puis, parmi tous les apprenants disponibles, filtrer ceux qui ont vu la vidéo, puis les compter. Ensuite, compter le nombre total d'apprenants. Et pour finir diviser le nombre d'apprenants ayant vu la vidéo par le nombre total d'apprenants."

Pour réaliser ce processus, que l'on peut exprimer en 5 étapes, il faut deux données nécessaires : le "temps" et "hasWatched?", un booléen indiquant si oui ou non l'utilisateur a vu la vidéo.

Essayez maintenant de créer ce processus. Pour ce faire, rechargez la page *"create_gap.html"* et recommencez les manipulations faites pour le processus précédent.

Questionnaire à remplir.

Feuille d'évaluation expérimentale - IFé- JJ/MM/AAAA

*Obligatoire

1. **Prénom :**

2. **Nom :**

3. **Métier** *

4. **Cas n° :** *

Données nécessaires

5. **Quelle est votre ressenti face à une plate-forme ne faisant qu'exprimer un processus d'analyse ?** *

6. **Avez-vous compris la notion de données nécessaires ?** *
   Une seule réponse possible.
   - ◯ Oui
   - ◯ Non

7. **Justifiez votre choix :**

8. **Pensez-vous que travailler avec les "données nécessaires" suffisent à exprimer un processus d'analyse ?** *
   Il ne s'agit ici que de l'expression, pas de l'instanciation.
   Une seule réponse possible.
   - ◯ Oui
   - ◯ Non

9. **Justifiez votre choix :**

Listes

10. **Avez-vous compris la notion de listes initiales ?** *
    Une seule réponse possible.
    - ◯ Oui
    - ◯ Non

11. **Justifiez votre choix :**

12. **Avez-vous compris la notion de listes calculées ?** *
    Une seule réponse possible.
    - ◯ Oui
    - ◯ Non

13. **Justifiez votre choix :**

14. **Avez-vous compris comment sont produites ces listes calculées ?** *
    Une seule réponse possible.
    - ◯ Oui
    - ◯ Non

15. **Justifiez votre choix :**

16. **Avez-vous compris comment utiliser ces listes calculées ?** *
Une seule réponse possible.
- ◯ Oui
- ◯ Non

17. **Justifiez votre choix :**

18. **Pensez-vous qu'ajouter de la sémantique aux listes pourrait être utile lors de la description d'un processus d'analyse ?** *
Une seule réponse possible.
- ◯ Oui
- ◯ Non

19. **Justifiez votre choix :**

Opérateurs

20. **La notion d'opérateur est-elle claire ?** *
Une seule réponse possible.
- ◯ Oui
- ◯ Non

21. **Justifiez votre choix :**

22. **La notion d'entrée d'un opérateur est-elle claire ?** *
Une seule réponse possible.
- ◯ Oui
- ◯ Non

23. **Justifiez votre choix :**

24. **La notion de sortie d'un opérateur est-elle claire ?** *
Une seule réponse possible.
- ◯ Oui
- ◯ Non

25. **Justifiez votre choix :**

26. **La notion de paramètrage de l'opérateur est-elle claire ?** *
Une seule réponse possible.
- ◯ Oui
- ◯ Non

27. **Justifiez votre choix :**

28. **Avez-vous été capable d'utiliser les bons opérateurs ?** *
Une seule réponse possible.

- ◯ Oui
- ◯ Non

29. **Justifiez votre choix :**

30. **Vous a-t-il manqué des opérateurs ?** *
Une seule réponse possible.

- ◯ Oui
- ◯ Non

31. **Justifiez votre choix :**

32. **Pensez-vous que tous les opérateurs que vous utilisez normalement puissent être représentés de cette manière ?** *
Une seule réponse possible.

- ◯ Oui
- ◯ Non

33. **Justifiez votre choix :**

Processus d'analyse

34. **Avez-vous réussi à exprimer le processus d'analyse pour ce cas ?** *
Une seule réponse possible.

- ◯ Oui
- ◯ Non

35. **Justifiez votre choix :**

36. **La difficulté de réalisation du processus d'analyse a été :** *
Une seule réponse possible.

- ◯ Très facile
- ◯ Facile
- ◯ Peu facile
- ◯ Pas facile
- ◯ Pas réussi

37. **Justifiez votre choix :**

38. **Avez-vous compris que le processus d'analyse était indépendant des plates-formes ?** *
Une seule réponse possible.

- ◯ Oui
- ◯ Non

39. **Justifiez votre choix :**

40. **Avez-vous compris que le prototype n'effectuait pas le processus d'analyse, mais le représentait uniquement ?** *
Une seule réponse possible.

- ◯ Oui
- ◯ Non

41. **Justifiez votre choix :**

42. **Identifiez-vous bien le processus d'analyse tel que représenté dans la 2ème zone (zone de description + graphe) ?** *
La zone de description est la zone où vous placez les opérateurs et les données nécessaires.
Une seule réponse possible.

- Oui
- Non

43. **Justifiez votre choix :**

44. **Pressentez-vous des limitations quant à cette approche ?** *
Une seule réponse possible.

- Oui
- Non

45. **Justifiez votre choix :**

Général - 1/2

46. **La notion d'étape vous a-elle parue assez claire ?** *
Une seule réponse possible.

- Oui
- Non

47. **Justifiez votre choix :**

48. **Le graphe de visualisation vous a-t-il aidé ?** *
Une seule réponse possible.

- Oui
- Non

49. **Justifiez votre choix :**

Général 2/2

50. **Voyez-vous l'intêret d'une telle approche ?** *
Une seule réponse possible.

- Oui
- Non

51. **Justifiez votre choix :**

52. **Êtes-vous intéressé par cette approche ?** *
Une seule réponse possible.

- Oui
- Non

53. **Justifiez votre choix :**

54. **Pensez-vous pouvoir exprimer tous vos processus de cette manière (en considérant un jeu plus important d'opérateurs) ? ***
Une seule réponse possible.

- ◯ Oui
- ◯ Non

55. **Justifiez votre choix :**

56. **Pensez-vous que cet outil peut servir d'alternative à une solution papier ? ***
Une seule réponse possible.

- ◯ Oui
- ◯ Non

57. **Justifiez votre choix :**

58. **Pensez-vous que cet outil peut permettre d'être plus explicite vis-à-vis du demandeur ? ***
Une seule réponse possible.

- ◯ Oui
- ◯ Non

59. **Justifiez votre choix :**

Commentaire général

60. **Remarques - Avis**

A = Analyste, PA = Processus d'Analyse, LI = Liste Initiale, LC = Liste Calculée, PTF = PlaTe-Forme

		Très facile	Facile	oyennement Fac	Pas facile	Pas réussis	
Étape préalable	**1 - Aisance globale de A dans sa tache de réflexion du PA**	☐	☐	☐	☐	☐	T01 + M01
	Commentaire :						
	2 - Création d'un PA sans données préalables	☐	☐	☐	☐	☐	T02 + M01
	Commentaire :						
	3 - Facilité à trouver un problème à traiter	☐	☐	☐	☐	☐	T02 + M01
	Commentaire :						
	4 - Facilité à associer un PA au problème	☐	☐	☐	☐	☐	T02 + M01
	Commentaire :						
	5 - Facilité à décrire le but du PA	☐	☐	☐	☐	☐	T02 + M01
	Commentaire :						
		Excellent	**Bon**	**Moyen**	**Passable**	**Médiocre**	
	6 - Précision de la description du but	☐	☐	☐	☐	☐	
	Commentaire :						
	7 - Faciliter à décrire la difficulté du PA	☐	☐	☐	☐	☐	T02 + M01
	Commentaire :						
	8 - Jugement de la complexité du PA selon A	☐	☐	☐	☐	☐	M01
	Commentaire :						
		Oui	**Non**				
	9 - Données préalables nécessaires pour A ?	☐	☐				
	Commentaire :						
	10 - Utilisation du processus d'exemple ?	☐	☐				
	Commentaire :						
	11 - But du PA concorde avec PA ?	☐	☐				
	Commentaire :						

		Excellent	Bon	Moyen	Passable	Médiocre	
	12 - Compréhension de A concernant l'outil	☐	☐	☐	☐	☐	T02 + C01 + C02 + E01 + E04
	Commentaire :						
	13 - Compréhension de A concernant l'indicateur recherché	☐	☐	☐	☐	☐	T02 + M01 + E01...4
	Commentaire :						
	14 - Appropriation de l'interface par A	☐	☐	☐	☐	☐	C01,2 + M01 + E01 + E02 +E04
	Commentaire :						
	15 - Aisance de création de la liste	☐	☐	☐	☐	☐	T02 + C02
	Commentaire :						
	16 - Aisance de création des éléments	☐	☐	☐	☐	☐	T02 + C02 + M01
	Commentaire :						
	17 - Utilisation des éléments de la LI	☐	☐	☐	☐	☐	C01 + M01 + E03

Familiarisation avec l'outil	Commentaire :						
	18 - Compréhension de A concernant les opérateurs affichés	☐	☐	☐	☐	☐	C01 + C02 + E01
	Commentaire :						
	19 - Aisance à l'identification du bon opérateur à utiliser	☐	☐	☐	☐	☐	T02 + C01 + C02
	Commentaire :						
	20 - Aisance de la sélection de l'opérateur	☐	☐	☐	☐	☐	T02 + C01 + C02 + E01 + R01
	Commentaire :						
	21 - Conscience de la complétion de l'étape une fois configurée	☐	☐	☐	☐	☐	C01 + E01 + E03 + E03
	Commentaire :						
	22 - Aisance concernant le paramétrage de l'opérateur	☐	☐	☐	☐	☐	C01 + C02 + E04
	Commentaire :						
	23 - Compréhension de la notion de LC	☐	☐	☐	☐	☐	T02 + C02 + E01 + E02 + E04
	Commentaire :						
	24 - Aisance quant à l'utilisation des éléments de la LC	☐	☐	☐	☐	☐	T02 + C01 + C02 + E01
	Commentaire :						
	25 - Autonomie de A sur la sélection de l'opérateur	☐	☐	☐	☐	☐	T02 + C01 + C02 + E01 + E02 + E03
	Commentaire :						
	26 - Qualité de la configuration du 2eme opérateur	☐	☐	☐	☐	☐	C01 + CO2 + E04
	Commentaire :						
	27 - Feedback apporté par le graphe à A	☐	☐	☐	☐	☐	C01
	Commentaire :						
	28 - Aisance, sur un plan cognitif, de A à sélectionner ses différentes étapes pour la sauvegarde	☐	☐	☐	☐	☐	C01 + C02 + E03 + E01
	Commentaire :						
	29 - Qualité général de la compréhension de A concernant cette partie	☐	☐	☐	☐	☐	ALL
	Commentaire :						
		Oui	**Non**				
	30 - A lit en entier avant de commencer ?	☐	☐				
	Commentaire :						
	31 - A a eu des questions lors de la compréhension de l'outil ?	☐	☐				
	Commentaire :						
	32 - Élement de LC correctement choisi ?	☐	☐				
	Commentaire :						
	33 - A a configuré le 2eme opérateur ?	☐	☐				
	Commentaire :						
	34 - A s'est servi du graphe ?	☐	☐				
	Commentaire :						

	35 - PA terminé ?	☐	☐	
	Commentaire :			
	36 - Qualité du PA	☐	☐	
	Commentaire :			
	37 - Temps de réalisation du PA	☐	☐	
	Commentaire :			
	38 - Intervention requise ?	☐	☐	
	Commentaire :			

		Excellent	**Bon**	**Moyen**	**Passable**	**Médiocre**	
Mise en oeuvre de votre processus d' analyse	**39 - Compréhension de la création des données initiales**	☐	☐	☐	☐	☐	T02 + M01 + C02 + E01 + E03 + E04 + E02
	Commentaire :						
	40 - Qualité de créations des données initiales	☐	☐	☐	☐	☐	E01 + E04 + E03 + E02
	Commentaire :						
	41 - Qualité des choix des opérateurs	☐	☐	☐	☐	☐	E01 + E02 + E03 + E04
	Commentaire :						
	42 - Compréhension du fonctionnement des opérateurs	☐	☐	☐	☐	☐	M01 + E01 + E02 + E03 + E04
	Commentaire :						
	43 - Configuration des opérateurs	☐	☐	☐	☐	☐	M01 + C01
	Commentaire :						
	44 - Utilisation des LC	☐	☐	☐	☐	☐	T02 + C02 + E01 + E02 + E04
	Commentaire :						
	45 - Manipulation de l'interface	☐	☐	☐	☐	☐	M01 + C01 + C02
	Commentaire :						
	46 - Compréhension de l'enchainement des étapes	☐	☐	☐	☐	☐	E03 + M01 + C01 + E01
	Commentaire :						
	47 - Organisation de A	☐	☐	☐	☐	☐	M01 + C01 + E03
	Commentaire :						
	48 - Qualité de l'export (A ne choisit que les étapes pertinentes)	☐	☐	☐	☐	☐	M01 + C01 + C02 + E01 + E03
	Commentaire :						
	49 - Autonomie générale	☐	☐	☐	☐	☐	T01 + M01 + C01 + E01 + E02 + E03 + E04 + R01
	Commentaire :						
	50 - Compréhension générale	☐	☐	☐	☐	☐	T01 + M01 + E01...4
	Commentaire :						
	51 - Facilité générale du PA (en terme de complexité)	☐	☐	☐	☐	☐	M01 + C01 + C02 + E03 + E04
	Commentaire :						
	52 - Facilité à exprimer le PA	☐	☐	☐	☐	☐	T01 + E01...4 + M01
	Commentaire :						
		Oui	**Non**				

53 - PA choisit identique à PA initiale ?	☐	☐
Commentaire :		
54 - Tout les éléments ont été définis la 1er fois par A ?	☐	☐
Commentaire :		
55 - Utilisation du graphe ?	☐	☐
Commentaire :		
56 - A cherche à réalisé son processus sur une PTF précise ?	☐	☐
Commentaire :		
57 - A possède des étapes non selectionnées à l'export ?	☐	☐
Commentaire :		
58 - A a supprimé des données ou des étapes ?	☐	☐
Commentaire :		
59 - PA final ~ PA initial ?	☐	☐
Commentaire :		
60 -PA terminé ?	☐	☐
Commentaire :		
61 - Qualité du PA	☐	☐
Commentaire :		
62 -Temps de réalisation du PA	☐	☐
Commentaire :		
63 - Intervention requise ?	☐	☐
Commentaire :		

	Excellent	**Bon**	**Moyen**	**Passable**	**Médiocre**
64 - Possibilité d'expression de l'outil actuel selon A	☐	☐	☐	☐	☐
Commentaire :					
65 - Aisance d'expression pour A	☐	☐	☐	☐	☐
Commentaire :					
66 - Compréhension de A concernant la notion de boîte noire avec inputs et outputs que sont les opérateurs	☐	☐	☐	☐	☐
Commentaire :					
67 - Compréhension de A sur le fait qu'un PA est une succession d'opérateurs	☐	☐	☐	☐	☐
Commentaire :					
68 - Conscience de A quant au fait que l'outil n'utilise que des concepts et pas les données	☐	☐	☐	☐	☐
Commentaire :					
69 - Clarté des relations entre les éléments	☐	☐	☐	☐	☐
Commentaire :					
70 - Clarté des relations entre les éléments et les listes	☐	☐	☐	☐	☐

Hypothèse	Commentaire :					
	71 - Clarté de la production des opérateurs (comment les LCs sont produites)	☐	☐	☐	☐	☐
	Commentaire :					
	72 - Clarté des LC	☐	☐	☐	☐	☐
	Commentaire :					
		Oui	**Non**			
	73 - A a rencontré des difficultés sans avoir les données directement ?	☐	☐			
	Commentaire :					
	74 - Besoin de feedback pour la configuration des opérateurs ?	☐	☐			
	Commentaire :					
	75 - Les problèmes d'interoperabilité ont-ils été considéré par A pendant le developpement ?	☐	☐			
	Commentaire :					
	76 - A s'est-il soucié de la PTF d'analyse ?	☐	☐			
	Commentaire :					
	77 - A a réussi à créer tout ce qu'il voulait ?	☐	☐			
	Commentaire :					

Documents liés à l'expérimentation des concepts de CAPTEN-ONION

E

E.1 Protocole

Protocole Expérimental

Septembre 2017

Introduction

L'objectif de l'expérimentation à laquelle vous participez est d'étudier les effets de la capitalisation de processus d'analyse de traces. En particulier la réutilisation d'analyses quand vous mettrez en oeuvre votre propre analyse.

L'expérimentation reposera sur des outils d'analyse classiques (type Weka, Excel…) qui sont mis à votre disposition, ainsi que sur notre prototype expérimental, nommé CAPTEN (Capitalization of Analysis Processes for Technology Enhanced learNing). Cet outil permet de conceptualiser les processus d'analyse de traces d'apprentissage et leurs composants ; de s'émanciper des contraintes techniques générées par les outils ; de structurer l'information véhiculée par ces processus grâce à une description narrative sur des concepts utilisés.

Compte tenu de la complexité apparente du prototype, la première partie de cette expérimentation est une partie explicative dudit prototype et des notions utilisées par le prototype. Elle vous présente les concepts principaux, les différentes pages, leur contenu et comment le lire. Cette première partie vous montre également le lien qui existe entre les concepts représentés dans les processus d'analyse et leur réification dans des outils d'analyses.

La deuxième partie vous fait intervenir directement en tant qu'analyste. Il vous sera proposé de répondre à deux besoins et vous pourrez choisir celui que vous souhaitez réaliser en premier (le deuxième étant idéalement à résoudre si le temps vous le permet).
Vous serez donc dans un premier temps autonome, pour résoudre comme vous l'entendez le besoin que vous aurez choisi, c'est-à-dire en créant une analyse dans un (ou plusieurs) outil(s) d'analyse. Si vous ne savez pas comment réaliser le processus dans les outils d'analyse, si vous bloquez ou une fois le processus terminé, CAPTEN sera alors mis à votre disposition. Vous pourrez alors vous en servir pour essayer de résoudre, critiquer, améliorer… votre processus.

Il est à noter que CAPTEN est encore en phase de développement, que des bugs peuvent encore subsister, que des temps de chargement assez importants peuvent survenir ou que des comportements étranges peuvent arriver. N'hésitez pas à appeler le responsable de séance pour éviter toute mauvaise manipulation.

Familiarisation avec le prototype CAPTEN

Présentation des concepts de CAPTEN

Ci-dessous le lexique des éléments principaux que vous serez amenés à rencontrer dans CAPTEN :

- **Narrated Analysis Process** / **Processus d'analyse narré** (NAP) : un processus d'analyse narré décrit une analyse réalisée, de manière conceptuelle : il est constitué d'opérations narrées (opérateurs et processus), d'éléments narratifs, présentés ci-après ;
- **Narrated Operator** / **Opérateur narré** (NOP) : un opérateur narré représente un concept d'opération à appliquer. Par exemple, un opérateur de régression linéaire, une addition...
- **Narrated Operation** / **Opération narrée** : une opération narrée peut être soit un opérateur narré, soit un processus d'analyse narré.
- **Graph of concepts** / **Graphe de concepts** (RGTE) : il s'agit d'un graphe de variables relationnelles. Il permet de décrire les variables présentes dans les traces et d'indiquer les relations qui existent entre elles : il s'agit en quelque sorte d'une ontologie de la trace. Ces graphes servent à représenter l'état des variables avant et après l'application des opérations narrées, mais aussi à offrir un environnement contrôlé autour d'un vocabulaire standardisé pour partager les informations.
    - **Concept** (or **Node**) / **Concept** : il s'agit des variables de la trace (e.g. le titre d'une colonne de CSV). Ainsi, le haut niveau conceptuel de CAPTEN s'établit sur les variables et non plus sur les valeurs/données atomiques de ces variables.
    - **Property** (or **Relation** or **Edge**) / **Propriété** (ou **Relation)** : il s'agit des liens qui existent entre les variables de la trace. Faire ressortir ces relations permet d'apporter des informations importantes sur l'analyse et son contexte.
- **Step** / **Étape** : C'est une étape lors de l'analyse. Elle est constituée d'une opération narrée (pouvant être configurée), du graphe de concepts sur lequel cette opération narrée est appliquée et du graphe résultant.
- **Narrative Element** / **Élément narratif** : il s'agit d'une information contrôlée, qui est sémantiquement définie. Elle est attachée à un élément de l'analyse par une relation.
- **Pattern** / **Patron** : l'effet d'une opération narrée sur un ensemble de concepts est définie par deux graphes de concepts, appelés patrons. L'un définit quels concepts sont attendus en entrée, l'autre représente quels concepts seront obtenus.
- **Relevant Concepts** / **Connaissances** : les connaissances sont des concepts, appartenant à un graphe de concepts, identifiés comme étant ce qui est recherché - le ou les objectifs - lors de l'analyse.

Le prototype que vous allez utiliser expérimente une méthode pour proposer aux analystes un moyen de capitaliser leurs processus d'analyse. Autrement dit, mettre à disposition pour la communauté ses processus d'analyse pour permettre leur reproductibilité, ainsi que de pouvoir les adapter pour les réutiliser dans des contextes différents, avec des outils d'analyse variés.

Pour ce faire, CAPTEN représente et met en relation les concepts véhiculés par les processus d'analyse implémentés dans les outils d'analyse. Cela permet de s'émanciper des contraintes techniques générées par les outils d'analyse sur les processus, et de se rapprocher de l'essence initiale des processus d'analyse tels qu'ils ont été pensés.
Cela implique donc que CAPTEN n'est pas un outil d'analyse conventionnel, en ce sens qu'il ne calcule pas les données comme pourrait le faire WEKA. Il permet de représenter le processus d'analyse, ses opérateurs, ses paramétrages… à un plus haut niveau.

Afin de structurer cette information de haut niveau, et permettre sa compréhension, nous avons opté pour une approche narrative dite contrôlée. Cela permet de documenter les processus d'analyse, en expliquant par exemple les choix d'implémentation, les hypothèses émises à chaque étape, etc., au travers de concepts sémantiques dédiés.

Présentation du prototype CAPTEN et réutilisation d'une analyse narrée

L'objectif de cette section est **(1)** de vous familiariser avec l'interface du prototype que vous serez amené-e à utiliser et **(2)** observer comment un processus d'analyse narré (et donc sa narration) peut être interprété pour permettre sa réutilisation.
Dans cette section, prenez bien le temps de suivre toutes les étapes de navigation présentées.
Si des difficultés devaient être rencontrées (compréhension, navigation), n'hésitez pas à contacter le responsable de séance.

1. Page d'accueil CAPTEN

Dans votre navigateur internet, une page internet à l'adresse http://localhost:3000/#/index devrait être ouverte (l'onglet s'appelle CAPTEN). Il s'agit de la page d'accueil du prototype.
La figure 1 représente la page d'accueil, où les points d'intérêts ont été identifiés par couleur. Ci-dessous, la signification de ces zones :

- (A) [En bleu] : Bandeau titre. La zone est cliquable pour remonter rapidement en haut de page.
- (B) [En cyan] : Zone d'onglets de navigation rapide. 4 zones sont présentes, mais dans le cadre de cette expérimentation, vous n'utiliserez que Index.
- (C) et (C') [En vert] : Bouton de retour arrière et avancer. Vous devez utiliser ces boutons pour aller sur les pages antérieures/postérieures que vous avez déjà visitées. Le code couleur des flèches indique sur quel type d'élément vous allez revenir (e.g. violet correspond à une étape). Vous **NE DEVEZ PAS** utiliser les boutons de navigation du navigateur.
- (D) [En jaune] : Boutons pour commencer à créer soit un processus d'analyse narré, soit un opérateur narré. Ils vous emmènent respectivement sur les pages http://localhost:3000/#/analysis/new et http://localhost:3000/#/nop/new

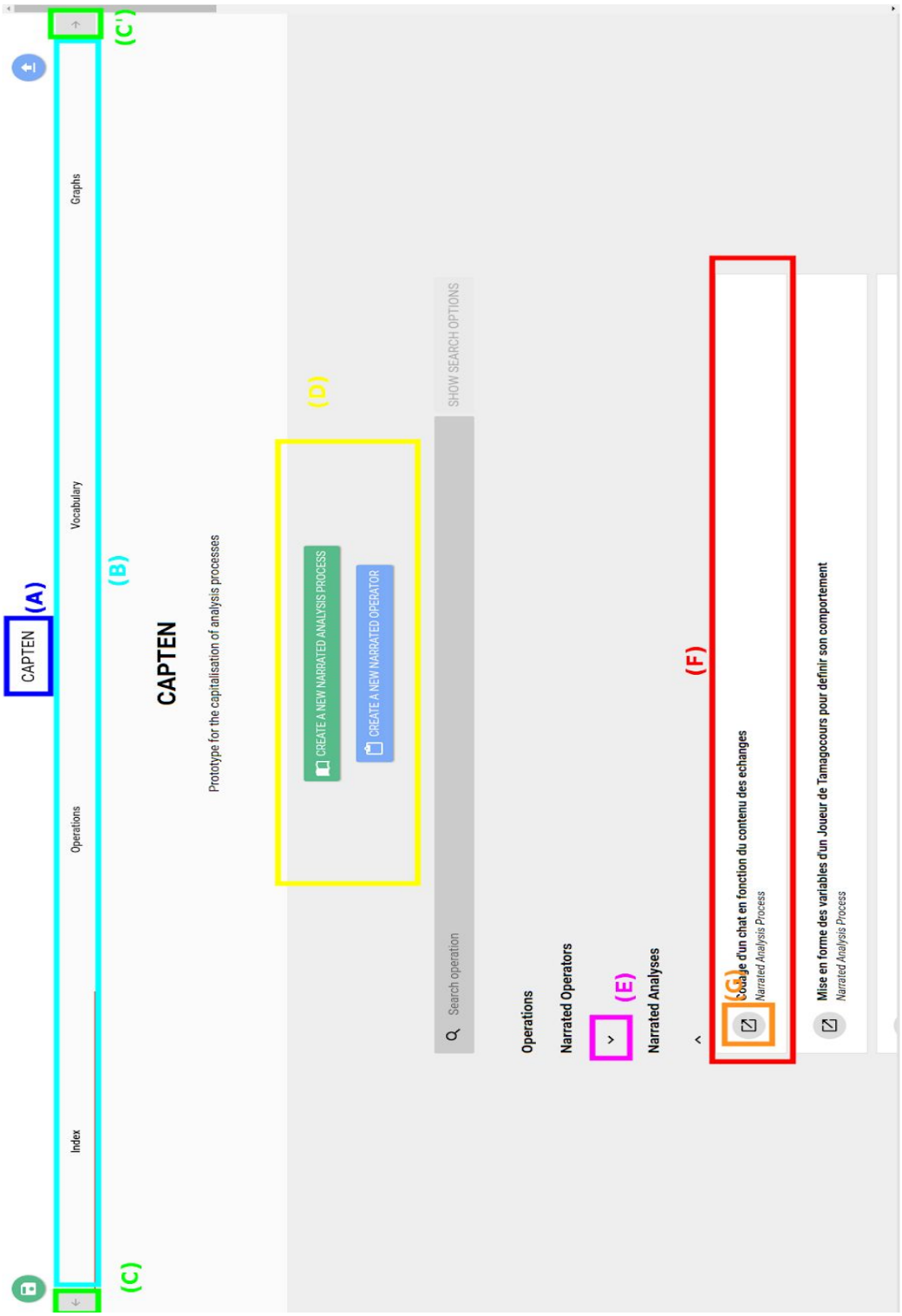


Figure 1. Page d'accueil de CAPTEN avec zones d'intérêt

- (E) [En rose] : Ces flèches permettent de dérouler la liste des opérations (NOP et NAP).
- (F) [En rouge] : Encart résumé d'une opération (ici, un processus d'analyse narré). Son nom est écrit en gras, et le type de l'opération en italique, en dessous.
- (G) [En orange] : Ce symbole représente l'ouverture de la page dédiée de l'objet en question. Ici, en cliquant sur cette icône, la page liée au processus d'analyse sera ouverte.

⚠ Vous ne devez **en aucun cas** réactualiser la page où vous vous trouvez. Toute la logique métier se trouve côté client. Si, par inadvertance, cela arrive, merci de contacter le responsable de séance.

2. Consulter un processus d'analyse narré

Nous allons désormais consulter un processus d'analyse narré simple et regarder comment il a été réalisé, quels sont ses objectifs, etc.

Toujours dans la page d'accueil, déroulez le menu des processus d'analyse narrés. Une liste d'encarts résumant les processus d'analyse narrés apparaît.
Cherchez le processus d'analyse narré intitulé "*Note Etudiant QCM*", puis cliquez sur l'encart résumé.
En cliquant sur l'encart résumé, ce dernier s'est agrandi et le bouton d'ouverture (G) est devenu bleu.

Cliquez sur ce bouton. Vous arriverez alors sur la page du processus d'analyse narré "*Note Etudiant QCM*".

⚠ Si vous n'avez aucun processus d'analyse narré, appelez votre responsable de séance.

3. Savoir lire et comprendre la page de processus d'analyse narré

La page de processus d'analyse narré est séparé en deux grandes parties, distinguées par leur couleur de fond (l'une en gris foncé, l'autre en gris clair).

La première partie, en gris foncé, compile toutes les informations rattachées directement au processus d'analyse narré. C'est dans cette partie que vous pouvez ajouter au processus de nouveaux éléments narratifs.

La deuxième partie, en gris clair, est une vue détaillée du processus d'analyse narré. Elle permet d'avoir une compréhension globale du processus d'analyse narré, avant d'entrer dans les détails de ces étapes.

3.1. La partie informative

Dans cette zone se situent les informations liées au processus d'analyse narré et elles l'enrichissent (zone (H) de la Figure 2). Chaque encart dans (H) représente un concept sémantiquement pré-déterminé, comme par exemple l'objectif de l'analyse.
On peut ainsi y voir une description globale de la démarche, en 3eme position dans (H).

Cette approche descriptive est renforcée par la possibilité d'imbriquer de nouveaux éléments narratifs à ces informations décrivant le processus. De cette manière, on obtient une approche narrative, ou il est possible d'évoluer au sein même des éléments narratifs utilisés.

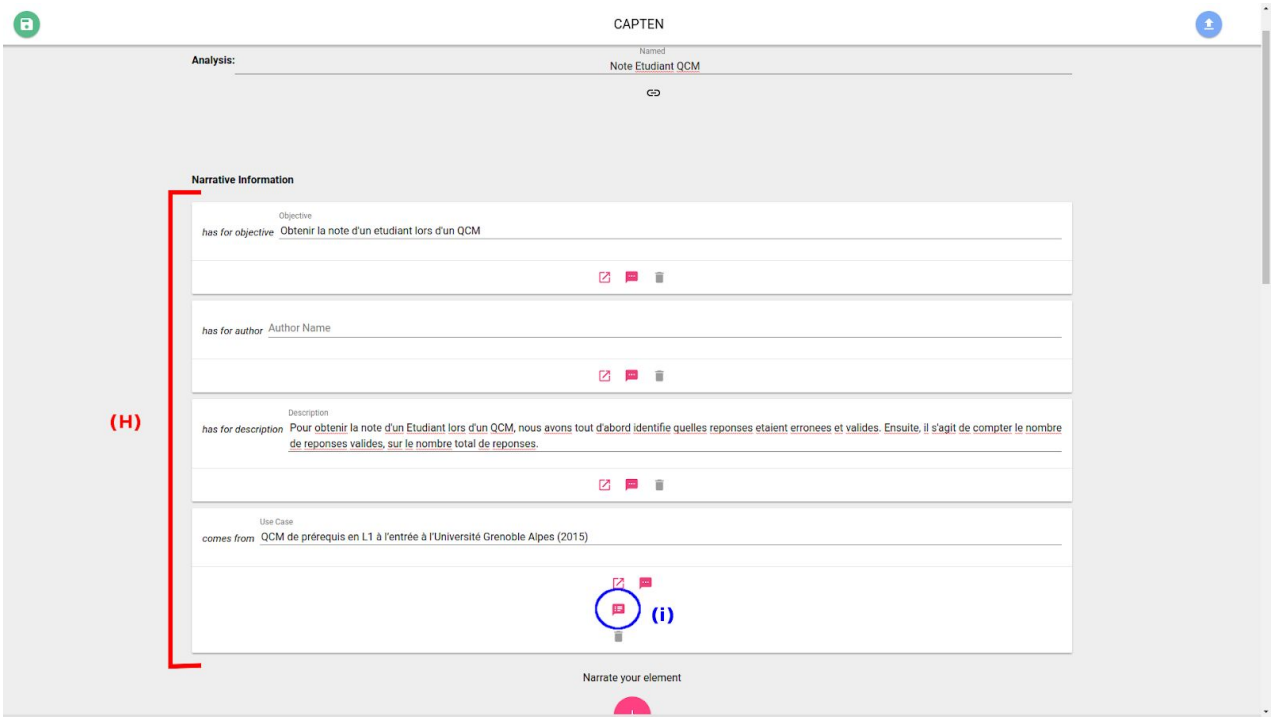


Figure 2. Partie narrative du processus d'analyse narré *Note Etudiant QCM*

Dans notre exemple, cliquez sur le bouton pour voir les éléments associés au "*Cas d'utilisation (Use Case)*". Il s'agit du bouton entouré en bleu, nommé (i).
En faisant cela, de nouveaux éléments narratifs apparaissent, qui documentent l'élément narratif de Cas d'utilisation. Comme par exemple le fait que le QCM provient du *LMS Chamillo*, ce qui vous apporte des informations contextuelles supplémentaires.

ℹ Les éléments narratifs sont donc utilisés pour apporter de l'information de manière structurée. On peut par exemple décrire les objectifs, ou bien les hypothèses sur lesquelles se fonde une analyse.

⚠ Notez la différence entre les deux icônes ci-dessous :

Celle qui permet d'obtenir les éléments narratifs liés à d'autres éléments narratifs est la deuxième **(2)** icône. C'est celle-ci qui vous intéresse dans cette expérimentation.
La première **(1)** icône permet de créer un nouvel élément narratif pour l'élément narratif en focus.

3.2. La vue détaillée

La vue détaillée d'un processus d'analyse narré est découpée verticalement en trois sous-parties.

La première sous-partie de cette vue, appelée *Initial concepts required*, représente quelles variables de la trace (mis en relations) ont été utilisées lors de l'analyse. Cela permet d'avoir une information rapide sur les prérequis et le contexte de l'analyse.
La Figure 3 montre ce à quoi ressemble cette mise en relation des variables pour le processus actuellement consulté.

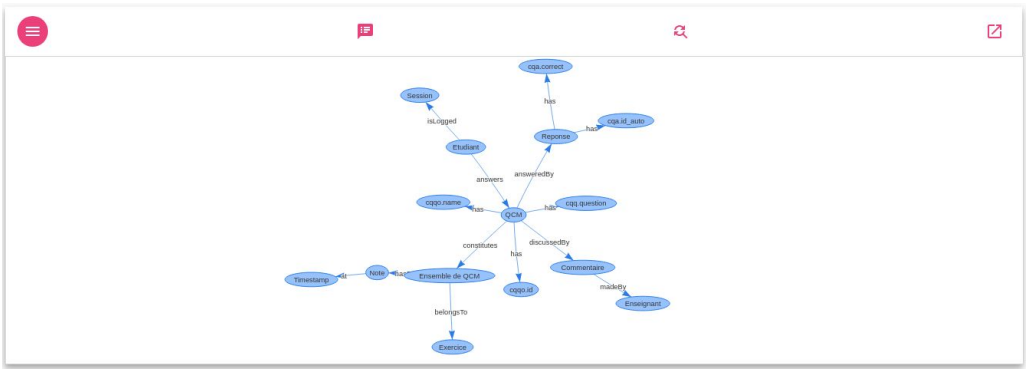


Figure 3. Graphe de concept initial du processus d'analyse narré *Note Etudiant QCM*

Cliquez sur la bulle de dialogue, en deuxième position dans la barre de menu (avant la loupe fléchée). Les éléments narratifs liés au graphe de concepts et ce qu'il représente apparaissent.
Cliquez de nouveau sur la bulle pour fermer la zone narrative.

i La loupe fléchée ne nous intéresse pas lors de cette expérimentation.

Cliquez sur la dernière icône. La page dédiée à ce graphe de concepts s'ouvre, avec le graphe en plus grand, des actions supplémentaires dans la barre de menu, ainsi que les éléments narratifs. Les nouveaux boutons ne nous intéressent pas.
Prenez un instant pour comprendre le graphe et les liens qui existent entre les concepts présents.

? Dans cet exemple, la personne qui a réifié son graphe de concept a utilisé des concepts non contrôlés, propres à son contexte, comme *cqa.id_auto.* Aucune documentation n'étant fournie, vous ne pouvez que supposer leur signification. Les relations (les flèches du graphe) peuvent

vous aider à mieux comprendre.

Revenez en arrière en utilisant la flèche retour arrière (C), comme montré sur la Figure 1. La couleur verte représente une page de type “processus d’analyse narré”.

La deuxième partie, *Steps overview*, donne un aperçu global des étapes du processus d’analyse narré, de leur nom/objectif. Cela permet de comprendre rapidement l’enchaînement d’actions effectuées au sein de l’analyse.

Figure 4. Vue globale des étapes réalisées au sein du processus d’analyse narré Note Etudiant QCM

Le symbole ✔ en haut à gauche des encarts indique que l’étape est complète. L’information *Position* indique quelle position occupe une étape de manière séquentielle.

Enfin, la troisième partie représente le workflow de l’analyse. Autrement dit, cette partie représente graphiquement comment les sorties des étapes précédentes sont utilisées dans les étapes suivantes.

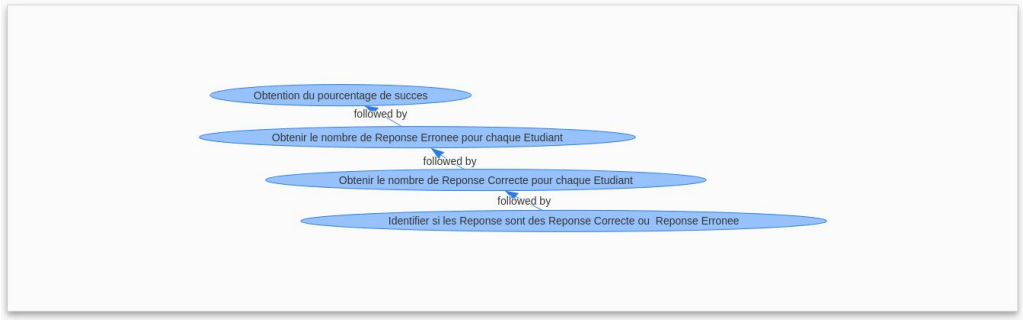


Figure 5. Flow d’analyse du processus d’analyse narré Note Etudiant QCM

Dans notre exemple simple, les sorties des étapes sont directement utilisées dans les étapes suivantes, créant un processus linéaire. Cependant, il est commun de voir apparaître des branches dans des processus plus complexes.
Cliquez, dans la zone *Steps Overview*, en bas à droite de l’encart de la première étape (*Identifier si les Reponse sont des Reponse Correcte ou Reponse Erronee*), sur l’icône de lancement. Vous arriverez sur la page de l’étape en question.

4. Comprendre la page d'une étape

Cette page est, elle aussi, séparée en deux zones. La première partie est, comme pour les processus d'analyse narrés, une zone narrative (gris foncé) ; la deuxième est une zone représentant ce qui est effectué pendant l'étape. La Figure 6 montre cette deuxième zone, assez complexe à comprendre.

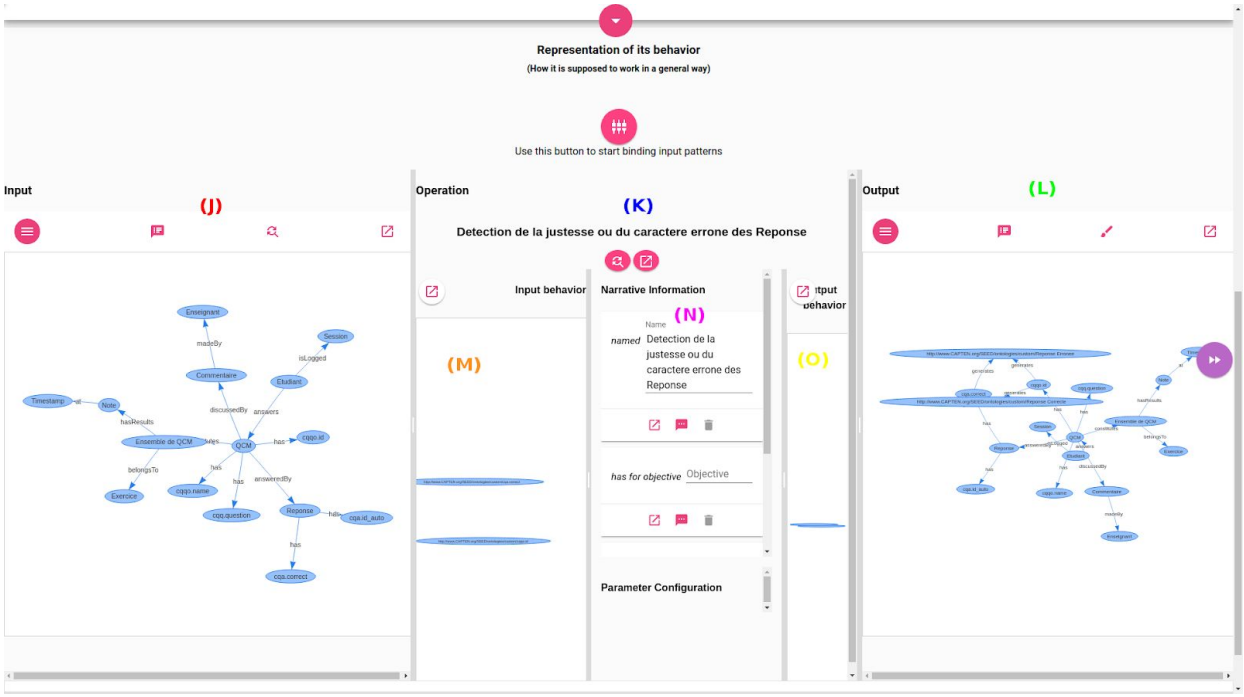


Figure 6. Définition comportementale de la première étape du processus d'analyse narré Note Etudiant QCM

La partie (J), à gauche, représente les variables de la trace qui ont sont utilisés par l'étape. Cette zone revient à dire que cette étape s'applique sur ce jeu de concepts, et va les modifier.

> ℹ Vous aurez sans doute remarqué que ce graphe de concepts est en réalité le graphe de concepts initial, vu plus tôt. En effet, il s'agit de la première étape de notre processus : il est donc normal que ce graphe initial soit utilisé en premier lieu.

La modification qui sera appliquée à ces concepts d'entrée est fonction de l'opération choisie, représentée dans la zone centrale, notée (K). Ici, par exemple, on applique une opération permettant de détecter si oui ou non les réponses sont correctes ou non.
Les concepts attendus pour une telle opération sont représentés dans la zone centrale gauche, notée (M). Ici, on attend deux concepts, respectivement cqa.correct et cqqo.id, pour que l'application de l'opération sur les concepts initiaux soit cohérente.

> ℹ Cette opération est un processus d'analyse défini préalablement par quelqu'un qui, ici, est

réutilisé. Vous pourrez vous renseigner en allant le consulter après cette introduction.

La zone centrale droite (O) représente ce que produit l'opération. Comme nous pouvons l'attendre d'une telle opération, elle génère un concept de réponse correcte et un concept de réponse erronée.

La zone centrale (N) regroupe à la fois les éléments narratifs de l'opération, et son paramétrage.

Enfin, la zone droite (L) est le graphe de concepts final. Il représente les concepts issus de l'opération appliquée sur les concepts initiaux.

i Pour résumer, lors de cette étape, on représente le fait d'utiliser une opération définie par l'utilisateur pour obtenir quelles réponses sont correctes ou non. Le patron d'entrée de cette opération attend deux concepts prédéfinis, et est appliqué sur le graphe de concepts initial. En sortie est représenté l'état des concepts, qui indique si la réponse est correcte ou non.

Cliquez sur le bouton avec la double flèche violette, tout à droite de l'écran, pour accéder à l'étape qui suit directement l'étape courante. Vous arrivez alors à l'étape *Obtenir le nombre de Reponse Correcte pour chaque Etudiant.* De la même manière que décrit précédemment, on représente le fait de vouloir compter le nombre de bonnes réponses pour, à la fin, obtenir une quantité de bonne réponse.

Cliquez encore une fois sur le bouton avec la double flèche. Cette fois-ci, l'étape se concentre sur compter le nombre d'échecs.

Pour conclure, cliquez une dernière fois sur le bouton avec la double flèche. Il s'agit de l'étape indiquant comment obtenir la proportion recherchée lors de l'analyse.
Le point intéressant ici est le concept identifié comme "connaissance" dans le graphe de sortie de cette dernière étape, en vert (cf. Figure 7).
D'après ce graphe, et en accord avec l'objectif du processus d'analyse narré, on voit que l'on cherchait à obtenir une proportion de succès pour l'apprenant.

5. Un point rapide sur les opérateurs narrés

Un opérateur narré permet donc de représenter des opérations atomiques/simples, ou bien d'indiquer qu'il existe une analyse qui prend tel concept en entrée et qui, en sortie, produira tel concept.

La page d'un opérateur est là aussi séparée en deux. La première partie est la partie liée à la narration.
La deuxième partie permet d'exprimer son patron d'entrée (sur quels concepts appliquer cette opération semble être cohérent), quels concepts de sortie sont attendus. Ces deux patrons sont exprimés sous forme de graphes.

Enfin, on exprime également quels patrons de paramétrage sont attendus. Ici sous forme textuelle avec la possibilité d'utiliser des éléments déjà définis par les utilisateurs. La Figure 8 montre cela.

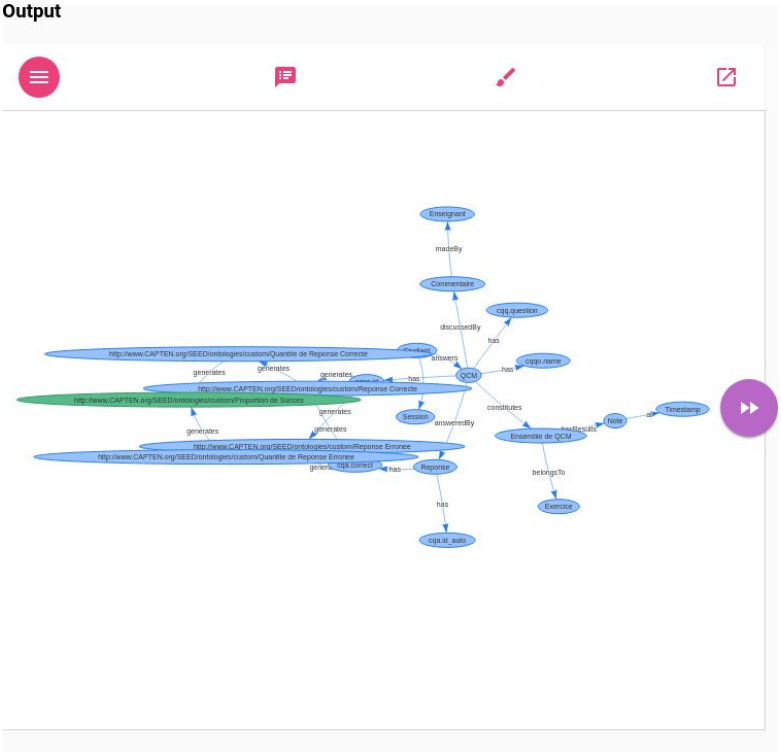


Figure 7. Graphe final du processus d'analyse narré "Note Etudiant QCM" où est visible le concept recherché, à savoir la proportion de succès.

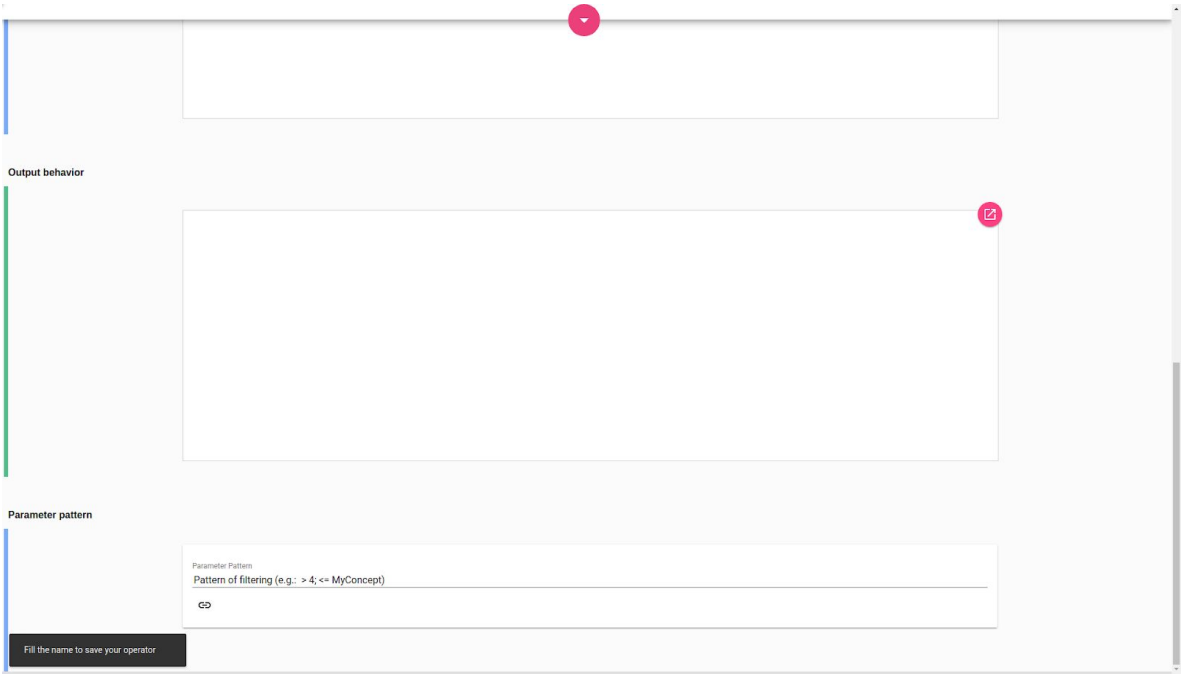


Figure 8. Partie basse de la deuxième zone d'un opérateur narré

6. Réutiliser le processus d'analyse narré

Dans cette sous section, nous allons observer comment les éléments narratifs peuvent être consultés pour vous aider à implémenter (en partie) le processus d'analyse narré *Note Etudiant QCM* dans un outil d'analyse.

Comme vous l'avez vu, la représentation d'un processus d'analyse narré est séparé en étapes atomiques.
Ainsi, à chaque étape, vous avez une représentation formelle de ce qui doit être fait (application d'une opération) pour faire évoluer l'état de vos variables (passer du graphe d'entrée au graphe de sortie). Cette représentation formelle vous permet de choisir ce qui doit être fait, et de contrôler comment vos variables évoluent.

Cependant, il se peut que vous ayez besoin d'adapter l'opération narrée à l'outil d'analyse désiré.

Là encore, les éléments narratifs constituent des renforts à la réutilisation. Certains éléments narratifs sont relatifs à l'implémentation dans un outil d'analyse et à certaines de ses particularités associées. Cela vous permet de savoir quel(s) opérateur(s) peut(vent) être utilisé(s) et comment ils pourrai(en)t être configuré(s).

⚠ Veuillez noter que les éléments narratifs en lien avec l'implémentation des processus d'analyse narrés, des opérateurs narrés, des étapes… ne sont pas encore implémentés en tant que tel actuellement.
À la place, référez-vous aux balises [Implementation] et [Implementation Example] des éléments narratifs de type Description.

Dans notre exemple, si nous nous considérons la première étape de *Note Etudiant QCM*, il s'avère en réalité que l'opération utilisée est un autre processus d'analyse.
Cherchez dans la liste des processus d'analyse narrés *Detection de la justesse ou du caractere errone des Reponse* et ouvrez-le.

Dans la zone centrale de l'opérateur *Conditional Enrichment*, vous pouvez distinguer des informations sur l'implémentation de cet opérateur au sein d'un outil d'analyse (ici, Undertracks). La Figure 9 montre tous les éléments narratifs liés à l'implémentation dans Undertracks de l'opération en question.
Déroulez les éléments narratifs liés à l'implémentation dans Undertracks pour en apprendre davantage, comme de quel opérateur il s'agit dans cet outil, et comment l'utiliser.

La Figure 10 montre l'utilisation de cet opérateur au sein d'UnderTracks, en suivant les informations présentes. C'est à vous - pour le moment - d'aller chercher l'opérateur dans l'outil de votre choix, et de le configurer. Les informations narrées vous assistent pour savoir si vous êtes encore dans le cadre de l'analyse.

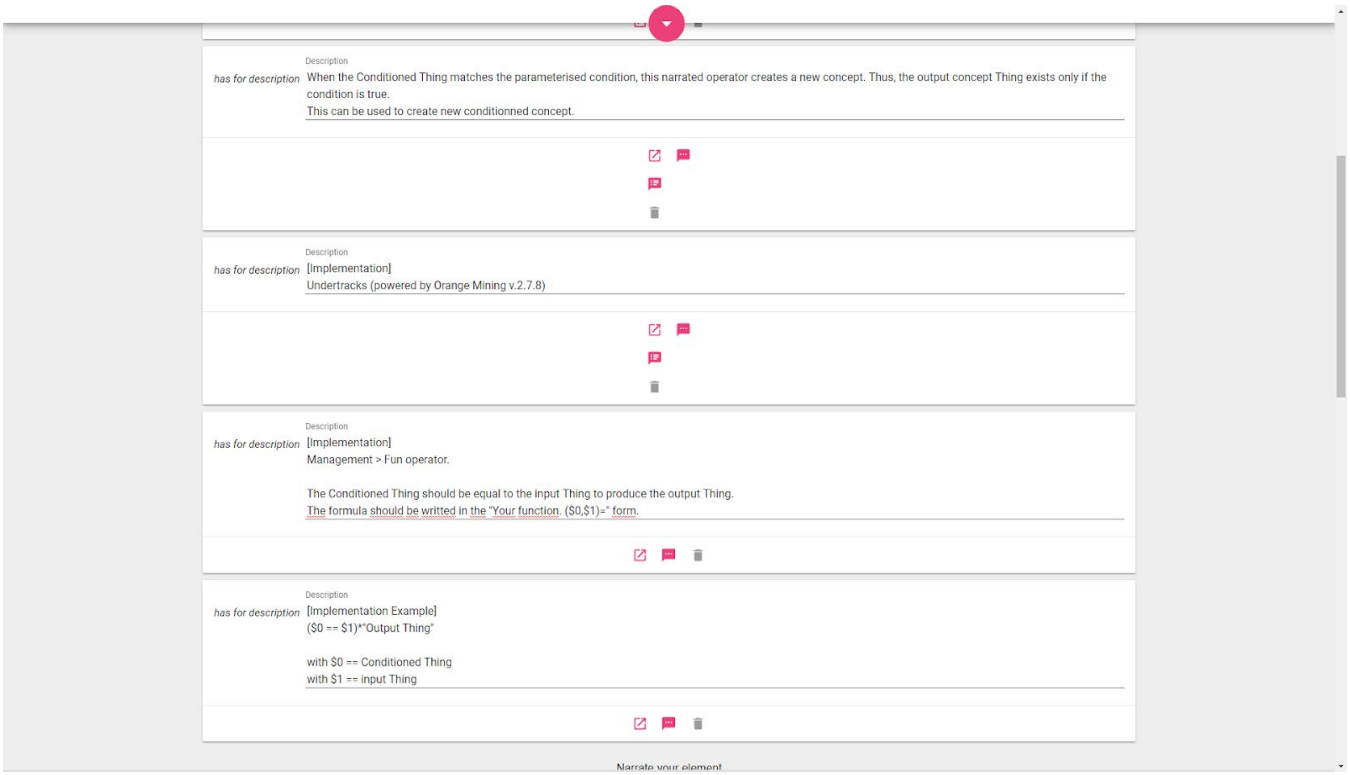


Figure 9. Éléments narratifs dédiés à l'implémentation avec Undertracks de l'opération d'enrichissement conditionnel.

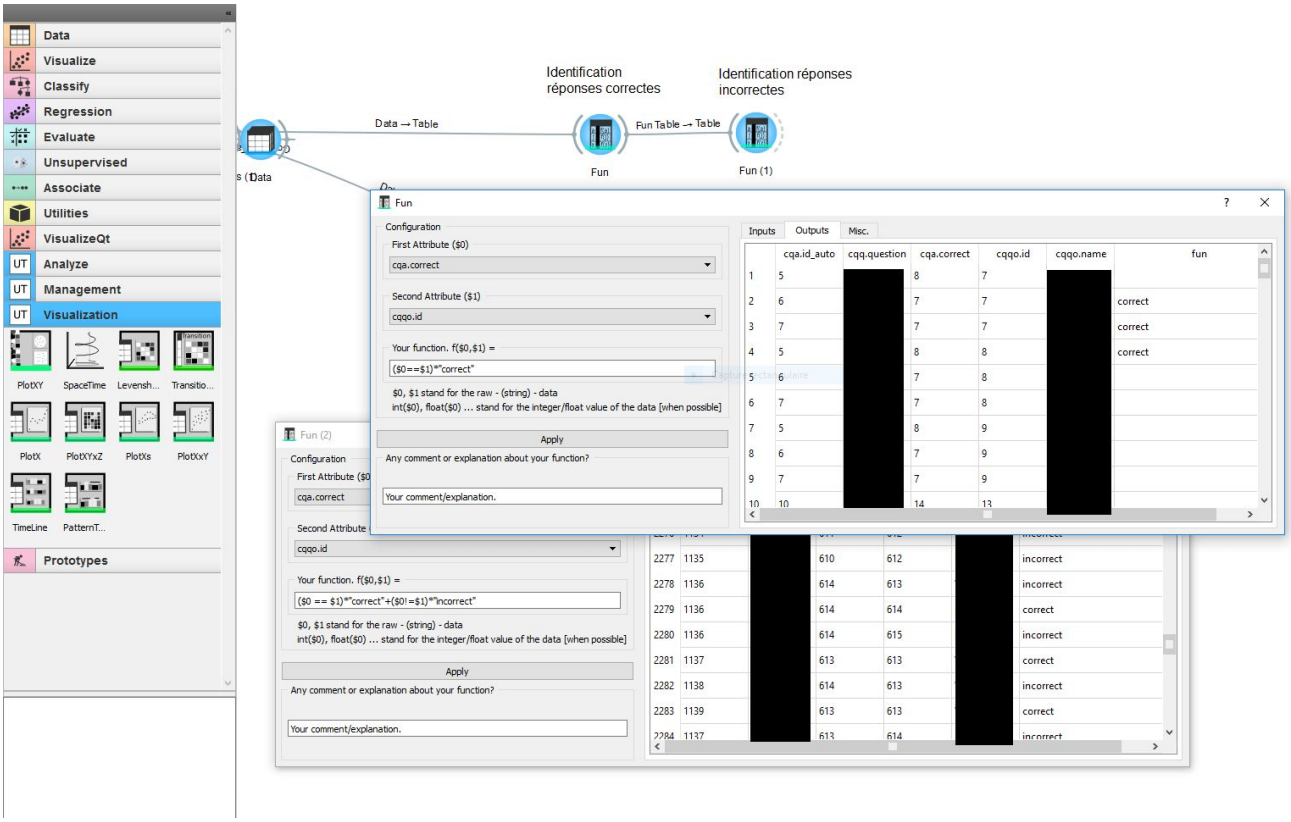


Figure 10. Réutilisation du processus d'analyse narré *Detection de la justesse ou du caractere errone des Reponse* dans Undertracks. La fenêtre en premier plan montre l'utilisation de l'opérateur Fun en suivant l'exemple d'implémentation fournie par les éléments narratifs. La fenêtre en second plan montre l'agrégation des deux étapes du processus d'analyse narré en une seule, en combinant les clauses puisque c'est possible avec Undertracks.

Expérimentation

Introduction

Vous endossez désormais le rôle d'analyste.
Votre client désire découvrir des informations pertinentes concernant les apprenants impliqués dans différents cours de MOOC, qui sont dispensés chaque semestre (automne et printemps).

Les traces d'apprentissage récoltées sur ces dispositifs d'apprentissage sont mises à votre disposition dans un format csv. Elles sont disponibles en ligne ici. Vous pouvez d'ores et déjà les consulter.
En outre, la notice concernant les variables contenues dans ces traces vous est fournie. La notice est accessible ici.

Enfin, voici les deux besoins qui vous sont adressés :

- **Être en mesure de savoir si un apprenant va ou non être certifié à la fin du cours** ;
- **Identifier des types d'apprenants, et si possible, par cours**.

Objectif

Choisissez **l'un** des deux besoins présentés ci-dessus que vous tenterez de résoudre.
Durant cette étape de résolution, **CAPTEN ne vous sera pas accessible**.
Cependant, vous êtes en autonomie complète pour résoudre le besoin que vous avez choisi. Vous pouvez donc utiliser les ressources que vous désirez pour vous documenter, vous inspirer…, afin de résoudre ce besoin.

Différents outils d'analyse sont disponibles sur le PC que vous utilisez (Undertracks/Orange, R, Knime, Weka, Libre Office Calc, Excel…). Utilisez celui ou ceux que vous voulez. Si jamais certains outils que vous avez l'habitude d'utiliser vous manquent, n'hésitez pas à le préciser. De plus, n'oubliez pas de l'indiquer à la fin de l'expérimentation dans le questionnaire, dans le champ correspondant.

Une fois que vous considérez votre analyse terminée, avertissez le responsable de séance. Il vous donnera alors la possibilité d'utiliser **CAPTEN** pour consulter les processus d'analyse narrés présents, et ainsi avoir un support supplémentaire pour vous inspirer et vous documenter. Vous pouvez réviser votre analyse au besoin.

> **?** Rappel : Les processus d'analyse narrés représentent des analyses réalisées dans des contextes plus ou moins génériques. Il est donc fort probable que le besoin exact que vous essayez de traiter n'existe pas. Cependant, certaines analyses décrites dans CAPTEN (ou sous-parties, connaissances, étapes, éléments narratifs…) pourraient s'adapter ou s'avérer pertinents pour votre situation.

Le cas échéant, n'oubliez pas de noter les processus d'analyse narrés que vous utiliserez, ainsi que les éventuelles modifications opérées lors de l'implémentation.

ℹ Il n'y a pas de souci si vous ne savez pas du tout comment répondre à un besoin, ou même aux deux. Simplement, manifestez-vous auprès du responsable de séance : il vous permettra alors d'utiliser **CAPTEN**. Si, même après cela, vous ne vous sentez toujours pas en mesure de répondre au besoin, passez à la partie "2nd Objectif".

ℹ Si vous bloquez lors de la réalisation de l'analyse et que vous ne pensez plus pouvoir continuer, manifestez-vous auprès du responsable de séance : il vous permettra alors d'utiliser **CAPTEN**. Si, même après cela, vous n'arrivez toujours pas à continuer, passez à la partie "2nd Objectif".

⚠ N'oubliez pas de sauvegarder régulièrement votre progression dans les différents outils utilisés.

2nd Objectif

Après avoir répondu au premier besoin, et après avoir averti le responsable de séance, vous pouvez commencer à répondre au deuxième besoin, si le temps le permet (le responsable de séance vous dira si c'est possible ou non).

Vous utiliserez la même démarche qu'avec le premier besoin. À savoir, là encore, que vous **ne devez pas utiliser CAPTEN**, mais que vous êtes en autonomie pour la réalisation de l'analyse.

Une fois l'analyse terminée, ou si vous ne savez pas comment faire, ou si vous vous retrouvez bloqué, signalez-le au responsable de séance pour qu'il vous donne accès à CAPTEN.

Fin et Questionnaire

Avant de terminer cette séance d'expérimentation, un questionnaire est à remplir ici.

Un grand merci pour votre temps et vos efforts ! :)

Feuille d'évaluation expérimentale

*Obligatoire

1. **Prénom** *

2. **Nom** *

3. **Profession** *

4. **Niveau d'expertise dans l'analyse des traces** *
   Une seule réponse possible.

	0	1	2	3	4	5	6	7	8	9	10	
Aucune connaissance	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Expert du domaine

XP - Compréhension des traces mis à disposition

5. **Quelle a été votre compréhension des traces de MOOC mis à votre disposition ?** *
   Une seule réponse possible.

	0	1	2	3	4	5	
Incomprises	◯	◯	◯	◯	◯	◯	Totalement comprises

6. **Justifiez votre note (quels sont les éléments présents qui vous ont aidé à comprendre ? Quels sont ceux qui manquaient ?)**

7. **Cette compréhension a-t-elle évolué au cours de l'expérimentation ?** *
   Une seule réponse possible.

   ◯ Oui *Passez à la question 9.*
   ◯ Non *Passez à la question 8.*

XP - Compréhension des traces

8. **Votre compréhension initiale des traces a-t-elle été aidée par les informations contenues dans CAPTEN ?** *
   Une seule réponse possible.

   ◯ Oui *Passez à la question 10.*
   ◯ Non *Passez à la question 13.*

XP - Compréhension des traces - 2

9. **Votre compréhension des traces s'est-elle améliorée grâce à l'utilisation du prototype CAPTEN ?** *
   Une seule réponse possible.

   ◯ Oui *Passez à la question 10.*
   ◯ Non *Passez à la question 13.*

XP - Compréhension des traces - Thx CAPTEN

10. **Quel a été l'impact de CAPTEN sur votre compréhension des traces** *
    Une seule réponse possible.

	0	1	2	3	4	5	
Inexistant	◯	◯	◯	◯	◯	◯	Complet

11. **Quels ont été les éléments qui vous ont aidé ?** *
    Plusieurs réponses possibles.

    ☐ Processus d'analyse narré
    ☐ Opérateur narré
    ☐ Concept
    ☐ Graphe de concepts
    ☐ Étape
    ☐ Patron
    ☐ Connaissances
    ☐ Élément narratif
    ☐ Autre :

12. **Pourquoi ?**

Passez à la question 14.

XP - Compréhension des traces identiques

13. **Vous a-t-il manqué quelque chose pour renforcer votre compréhension des traces ? Si oui, qu'était-ce ?**

Expérimentation - Contexte global

14. **Quel a été votre compréhension du contexte de l'analyse ?** *
Une seule réponse possible.

	0	1	2	3	4	5	
Incompris	◯	◯	◯	◯	◯	◯	Totalement compris

15. **Quels éléments (présents et non présents) vous ont aidé à donner cette note ?**

16. **Votre compréhension du contexte a-t-elle évolué au cours de l'expérimentation** *
Une seule réponse possible.

- ◯ Oui *Passez à la question 18.*
- ◯ Non *Passez à la question 17.*

XP - Compréhension du contexte

17. **Votre compréhension initiale a-t-elle été aidée par CAPTEN ?** *
Une seule réponse possible.

- ◯ Oui *Passez à la question 19.*
- ◯ Non *Passez à la question 22.*

XP - Compréhension du contexte - 2

18. **Votre compréhension s'est-elle améliorée grâce à l'utilisation du prototype CAPTEN ?** *
Une seule réponse possible.

- ◯ Oui *Passez à la question 19.*
- ◯ Non *Passez à la question 22.*

XP - Compréhension du contexte - Thx CAPTEN

19. **CAPTEN vous a-t-il aidé à mieux comprendre le contexte de l'analyse ?** *
Une seule réponse possible.

	0	1	2	3	4	5	
Absolument pas	◯	◯	◯	◯	◯	◯	Totalement

20. **Quels ont été les éléments qui vous ont aidé ?** *
Plusieurs réponses possibles.

- ☐ Processus d'analyse narré
- ☐ Opérateur narré
- ☐ Concept
- ☐ Graphe de concepts
- ☐ Étape
- ☐ Patron
- ☐ Connaissance
- ☐ Élément narratif
- ☐ Autre :

21. **Pourquoi ?**

Passez à la question 23.

XP - Compréhension du contexte - Fin

22. **Vous a-t-il manqué des informations pour mieux comprendre le contexte de l'analyse ? Si oui, lesquelles ?**

Expérimentation - Besoins

23. **À combien de besoins avez-vous su répondre** *
Une seule réponse possible.

0	1	2
◯	◯	◯

24. **Quel besoin avez-vous choisi en premier ?** *
Une seule réponse possible.

- ◯ Être en mesure de savoir si un apprenant va ou non être certifié à la fin du cours *Après avoir répondu à la dernière question de cette section, passez à la question 26.*
- ◯ Identifier des types d'apprenants, et si possible, par cours *Après avoir répondu à la dernière question de cette section, passez à la question 26.*
- ◯ Aucun des deux *Après avoir répondu à la dernière question de cette section, passez à la question 88.*

25. **Pourquoi avoir opté pour ce besoin en premier ?** *

_____
_____
_____
_____
_____

1er Besoin

26. **Quel a été votre niveau de compréhension de ce besoin ?** *
Une seule réponse possible.

	0	1	2	3	4	5	
Incompris	◯	◯	◯	◯	◯	◯	Totalement compris

27. **Aviez-vous une idée de ce qui était recherché ? (Par exemple, obtenir des groupes d'étudiants en lien avec le critère de certification)** *
Une seule réponse possible.

- ◯ Oui
- ◯ Non

28. **Pourquoi ?**

_____
_____
_____
_____
_____

29. **Aviez-vous une idée de comment réaliser votre analyse ?** *
Une seule réponse possible.

- ◯ Oui
- ◯ Non

30. **Pourquoi ?**

_____
_____
_____
_____
_____

31. **Saviez-vous quel(s) outil(s) utiliser et comment ?** *
Une seule réponse possible.

- ◯ Oui
- ◯ Non

32. **Lesquels ?** *

_____

33. **Pourquoi ?**

_____
_____
_____
_____
_____

1er Besoin - Réalisation

34. **Avez-vous réussi à réaliser l'analyse ?** *
Une seule réponse possible.

- ◯ Je n'ai pas su quoi faire lorque j'étais en autonomie *Passez à la question 37.*
- ◯ J'ai rencontré des difficultés qui m'ont empêché de finir l'analyse lorsque j'étais en autonomie *Passez à la question 37.*
- ◯ J'ai réussi à terminer le processus d'analyse lorsque j'étais en autonomie *Passez à la question 35.*
- ◯ Autre : _____

1er Besoin - Réalisation sans CAPTEN

35. **Que pensiez-vous des résultats que vous avez obtenus avec votre analyse par rapport au besoin, avant d'utiliser CAPTEN ?** *
Une seule réponse possible.

	0	1	2	3	4	5	
Pas satisfaisant	◯	◯	◯	◯	◯	◯	Excellent

36. **Pourquoi cette note ?**

1er Besoin - Ressources utilisées

37. **Quel(s) type(s) de ressources avez vous consulté pour réaliser votre analyse en autonomie ?** *

38. **Pourquoi ?**

39. **Vous a-t-il manqué des ressources** *
Une seule réponse possible.

- ◯ Oui
- ◯ Non

40. **Si oui, de quelle sorte ?**

41. **De manière générale, les ressources consultées étaient-elles suffisament structurées pour vous aider ?** *
Une seule réponse possible.

	0	1	2	3	4	5	
Pas du tout utile	◯	◯	◯	◯	◯	◯	Totalement utile

1er Besoin - Avec CAPTEN

42. **Dans la deuxième phase, CAPTEN a-t-il été bénéfique à votre analyse ?** *
Une seule réponse possible.

- ◯ Non, j'étais toujours bloqué, même après avoir accès à CAPTEN
- ◯ Oui, j'ai pu finir l'analyse *Passez à la question 45.*
- ◯ Oui, j'ai pu améliorer mon analyse *Passez à la question 45.*
- ◯ Non, CAPTEN ne m'a rien apporté de plus *Passez à la question 44.*

1er Besoin - Toujours bloqué

43. **D'après vous, pourquoi n'avez vous pas réussi à décrire cette analyse ?** *

Passez à la question 45.

1er Besoin - Aucun Apport

44. **Pourquoi CAPTEN ne vous a rien apporté ?** *

Passez à la question 57.

1er Besoin - Apport de CAPTEN

45. **D'après vous, quel a été le degré d'assistance de CAPTEN dans votre analyse ?** *
Une seule réponse possible.

	0	1	2	3	4	5	6	7	8	9	10	
Inutile	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Totale

46. **Quelle a été la difficulté pour trouver les informations qui vous ont aidés ?** *
Une seule réponse possible.

	0	1	2	3	4	5	
Trivial	◯	◯	◯	◯	◯	◯	Complexe

47. **Avez-vous trouvé des processus d'analyse narrés pouvant vous aider à résoudre votre besoin ?** *
Une seule réponse possible.

- ◯ Oui *Passez à la question 49.*
- ◯ Non *Passez à la question 48.*

1er Besoin - Apport CAPTEN - Aucun NAP

48. **Pourquoi n'avez-vous trouvé aucun processus d'analyse narré ? ***

_____

_____

_____

_____

_____

Passez à la question 57.

1er Besoin - Apport CAPTEN - NAP et réutilisation

49. **Quels processus d'analyse narrés vous ont aidés ?**

_____

_____

_____

_____

_____

50. **Avez-vous réutilisé ces processus d'analyse narrés dans votre analyse ? ***
Une seule réponse possible.

- ◯ Oui, tous
- ◯ Oui, une partie
- ◯ Non
- ◯ Autre : _____

51. **Avez-vous su adapter ces processus d'analyse narrés pour les faire correspondre à votre besoin ? ***
Une seule réponse possible.

	0	1	2	3	4	5	
Absolument pas	◯	◯	◯	◯	◯	◯	Totalement

52. **Pensez-vous avoir réussi à réutiliser ces processus d'analyse narrés pour votre besoin ? ***
Une seule réponse possible.

	0	1	2	3	4	5	
Absolument pas	◯	◯	◯	◯	◯	◯	Totalement

53. **En moyenne, comment estimez-vous la similarité entre les variables de votre besoin et les variables des processus d'analyse narrés qui vous ont été utiles ? ***
Une seule réponse possible.

	0	1	2	3	4	5	
Totalement différent	◯	◯	◯	◯	◯	◯	Identique

54. **En moyenne, comment estimez-vous la similarité entre le contexte de votre besoin et le contexte des processus d'analyse narrés qui vous ont été utiles ? ***
Une seule réponse possible.

	0	1	2	3	4	5	
Totalement différent	◯	◯	◯	◯	◯	◯	Identique

55. **En moyenne, comment estimez-vous la similarité entre les objectifs de votre besoin et les objectifs (connaissances) des processus d'analyse narrés qui vous ont été utiles ? ***
Une seule réponse possible.

	0	1	2	3	4	5	
Totalement différent	◯	◯	◯	◯	◯	◯	Identique

56. **Consulter CAPTEN vous a permis de : ***
Plusieurs réponses possibles.

- ☐ Avoir une meilleure idée de ce qui était attendu
- ☐ Terminer votre analyse
- ☐ Vérifier des choix d'implémentation
- ☐ Revoir mes paramétrages
- ☐ Chercher de nouveaux résultats
- ☐ Élaborer de nouvelles classes pour les apprenants
- ☐ Avoir une autre méthode à laquelle me comparer
- ☐ Mieux comprendre les variables
- ☐ Mieux comprendre les interactions entre les variables
- ☐ Avoir une assistance à la réutilisation
- ☐ Autre : _____

2eme Besoin

57. **Avez-vous traité le deuxième besoin ? ***
Une seule réponse possible.

- ◯ Oui *Passez à la question 58.*
- ◯ Non *Passez à la question 88.*

2eme Besoin - Intro

58. **Quel a été votre niveau de compréhension de ce besoin ? ***
Une seule réponse possible.

	0	1	2	3	4	5	
Incompris	◯	◯	◯	◯	◯	◯	Totalement compris

59. **Aviez-vous une idée de ce qui était recherché ? (Par exemple, obtenir des groupes d'étudiants en lien avec le critère de certification) ***
Une seule réponse possible.

- ◯ Oui
- ◯ Non

60. **Pourquoi ?**

61. **Aviez-vous une idée de comment réaliser votre analyse ?** *
Une seule réponse possible.

- ◯ Oui
- ◯ Non

62. **Saviez-vous quel(s) outil(s) utiliser et comment ?** *
Une seule réponse possible.

- ◯ Oui
- ◯ Non

63. **Lesquels ?** *

64. **Pourquoi ?**

2eme Besoin - Réalisation

65. **Avez-vous réussi à réaliser l'analyse ?** *
Une seule réponse possible.

- ◯ Je n'ai pas su quoi faire lorque j'étais en autononomie *Passez à la question 68.*
- ◯ J'ai rencontré des difficultés qui m'ont empêché de finir l'analyse lorsque j'étais en autonomie *Passez à la question 68.*
- ◯ J'ai réussi à terminer le processus d'analyse lorsque j'étais en autonomie *Passez à la question 66.*
- ◯ Autre :

2eme Besoin - Réalisation sans CAPTEN

66. **Que pensiez-vous des résultats que vous avez obtenus avec votre analyse par rapport au besoin, avant d'utiliser CAPTEN ?** *
Une seule réponse possible.

	0	1	2	3	4	5	
Pas satisfaisant	◯	◯	◯	◯	◯	◯	Excellent

67. **Pourquoi cette note ?**

2eme Besoin - Ressources utilisées

68. **Quel(s) type(s) de ressources avez-vous consultées ?** *

69. **Pourquoi ?**

70. **Vous a-t-il manqué des ressources** *
Une seule réponse possible.

- ◯ Oui
- ◯ Non

71. **Si oui, de quelle sorte ?**

72. **De manière générale, les ressources consultées étaient-elles suffisament structurées pour vous aider ?** *
Une seule réponse possible.

	0	1	2	3	4	5	
Pas du tout utile	◯	◯	◯	◯	◯	◯	Totalement utile

2eme Besoin - Avec CAPTEN

73. **Dans la deuxième phase, CAPTEN a-t-il été bénéfique à votre analyse ?** *
Une seule réponse possible.

- ◯ Non, j'étais toujours bloqué, même après avoir accès à CAPTEN *Passez à la question 74.*
- ◯ Oui, j'ai pu finir l'analyse *Passez à la question 76.*
- ◯ Oui, j'ai pu améliorer mon analyse *Passez à la question 76.*
- ◯ Non, CAPTEN ne m'a rien apporté de plus *Passez à la question 76.*

2eme Besoin - Toujours bloqué

74. **D'après vous, pourquoi n'avez-vous pas réussi à décrire cette analyse ?** *

_____
_____
_____
_____
_____

Passez à la question 76.

2eme Besoin - Aucun Apport

75. **Pourquoi CAPTEN ne vous a rien apporté ?** *

_____
_____
_____
_____
_____

Passez à la question 88.

2eme Besoin - Apport de CAPTEN

76. **D'après vous, quel a été le degré d'assistance de CAPTEN dans votre analyse ?** *
Une seule réponse possible.

	0	1	2	3	4	5	6	7	8	9	10	
Inutile	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Totale

77. **Quelle a été la difficulté pour trouver les informations qui vous ont aidés ?** *
Une seule réponse possible.

	0	1	2	3	4	5	
Trivial	◯	◯	◯	◯	◯	◯	Complexe

78. **Avez-vous trouvé des processus d'analyse narrés pouvant vous aider à résoudre votre besoin ?** *
Une seule réponse possible.

- ◯ Oui *Passez à la question 80.*
- ◯ Non *Passez à la question 79.*

2eme Besoin - Apport CAPTEN - Aucun NAP

79. **Pourquoi n'avez-vous trouvé aucun processus d'analyse narré ?** *

_____
_____
_____
_____
_____

Passez à la question 88.

2eme Besoin - Apport CAPTEN - NAP et réutilisation

80. **Quels processus d'analyse narrés vous ont aidés ?** *

_____
_____
_____
_____
_____

81. **Avez-vous réutilisé ces processus d'analyse narrés dans votre analyse ?** *
Une seule réponse possible.

- ◯ Oui, tous
- ◯ Oui, une partie
- ◯ Non
- ◯ Autre : _____

82. **Avez-vous su adapter ces processus d'analyse narrés pour les faire correspondre à votre besoin ?** *
Une seule réponse possible.

	0	1	2	3	4	5	
Absolument pas	◯	◯	◯	◯	◯	◯	Totalement

83. **Pensez-vous avoir réussi à réutiliser ces processus d'analyse narrés pour votre besoin ?** *
Une seule réponse possible.

	0	1	2	3	4	5	
Absolument pas	◯	◯	◯	◯	◯	◯	Totalement

84. **En moyenne, comment estimez-vous la similarité entre les variables de votre besoin et les variables des processus d'analyse narrés qui vous ont été utiles ?** *
Une seule réponse possible.

	0	1	2	3	4	5	
Totalement différent	◯	◯	◯	◯	◯	◯	Identique

85. **En moyenne, comment estimez--vous la similarité entre le contexte de votre besoin et le contexte des processus d'analyse narrés qui vous ont été utiles ?** *
Une seule réponse possible.

	0	1	2	3	4	5	
Totalement différent	◯	◯	◯	◯	◯	◯	Identique

86. **En moyenne, comment estimez-vous la similarité entre les objectifs de votre besoin et les objectifs (connaissances) des processus d'analyse narrés qui vous ont été utiles ?** *
Une seule réponse possible.

	0	1	2	3	4	5	
Totalement différent	◯	◯	◯	◯	◯	◯	Identique

87. **Consulter CAPTEN vous a permis de :** *
Plusieurs réponses possibles.

- ☐ Avoir une meilleur idée de ce qui etait attendu
- ☐ Terminer votre analyse
- ☐ Vérifier des choix d'implémentation
- ☐ Revoir mes paramétrages
- ☐ Chercher de nouveaux résultats
- ☐ Élaborer de nouvelles classes pour les apprenants
- ☐ Avoir une autre méthode à laquelle me comparer
- ☐ Mieux comprendre les variables
- ☐ Mieux comprendre les interactions entre les variables
- ☐ Avoir une assistance à la réutilisation
- ☐ Autre : _____

CAPTEN - Théorie

88. **Avez-vous compris la théorie derrière CAPTEN (processus d'analyse narrés, graphe de concepts, etc...)** *
Une seule réponse possible.

- ◯ Oui *Passez à la question 103.*
- ◯ Non *Passez à la question 89.*

Concepts de l'approche

89. **Pensez-vous avoir compris ce qu'est un processus d'analyse narré ?** *
Une seule réponse possible.

	0	1	2	3	4	5	
Pas du tout	◯	◯	◯	◯	◯	◯	Totalement

90. **Justifiez votre note**

_____

_____

_____

_____

_____

91. **Pensez-vous avoir compris ce qu'est un opérateur narré ?** *
Une seule réponse possible.

	0	1	2	3	4	5	
Pas du tout	◯	◯	◯	◯	◯	◯	Totalement

92. **Justifiez votre note**

_____

_____

_____

_____

_____

93. **Pensez-vous avoir compris ce qu'est un concept ?** *
Une seule réponse possible.

	0	1	2	3	4	5	
Pas du tout	◯	◯	◯	◯	◯	◯	Totalement

94. **Justifiez votre note**

_____

_____

_____

_____

_____

95. **Pensez-vous avoir compris ce qu'est un graphe de concepts ?** *
Une seule réponse possible.

	0	1	2	3	4	5	
Pas du tout	◯	◯	◯	◯	◯	◯	Totalement

96. **Justifiez votre note**

_____

_____

_____

_____

_____

97. **Pensez-vous avoir compris ce qu'est une étape ?** *
Une seule réponse possible.

	0	1	2	3	4	5	
Pas du tout	◯	◯	◯	◯	◯	◯	Totalement

98. **Justifiez votre note**

99. **Pensez-vous avoir compris ce qu'est la narration ?** *
Une seule réponse possible.

	0	1	2	3	4	5	
Pas du tout	◯	◯	◯	◯	◯	◯	Totalement

100. **Justifiez votre note**

101. **Pensez-vous avoir compris ce qu'est un patron ?** *
Une seule réponse possible.

	0	1	2	3	4	5	
Pas du tout	◯	◯	◯	◯	◯	◯	Totalement

102. **Justifiez votre note**

Général - 1/2

103. **Vous a-t-il manqué des éléments narratifs ?**
Une seule réponse possible.

- ◯ Oui
- ◯ Non

104. **Si oui, lesquels ?**

105. **Avez-vous utilisé CAPTEN pour rechercher des informations bien précises ?**
Une seule réponse possible.

- ◯ Oui
- ◯ Non

106. **Si oui, lesquelles ?**

Général - 2 / 2

107. **D'une manière générale, quantifiez l'apport de CAPTEN comme assistance à la réutilisation et l'adaptation de l'existant** *
Une seule réponse possible.

	0	1	2	3	4	5	6	7	8	9	10	
Aucun apport	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Indispensable

108. **La structuration de l'information dans CAPTEN vous a-t-elle aidé par rapport à des moyens plus conventionnels ?** *
Une seule réponse possible.

	0	1	2	3	4	5	6	7	8	9	10	
Aucunement	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Totalement

109. **Par rapport à une approche textuelle classique, quantifiez l'apport que CAPTEN a eu pour vous** *
Une seule réponse possible.

	0	1	2	3	4	5	6	7	8	9	10	
Aucun apport	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Indispensable

110. **Cochez cette case si vous ne savez pas quantifier l'apport de CAPTEN par rapport à une approche textuelle classique**
Une seule réponse possible.

- ◯ Ne sais pas quantifier l'apport de CAPTEN sur une approche textuelle

111. **Par rapport à une approche par workflow classique, quantifiez l'apport que CAPTEN a eu pour vous** *

Une seule réponse possible.

	0	1	2	3	4	5	6	7	8	9	10	
Aucun apport	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Indispensable

112. **Cochez cette case si vous ne savez pas quantifier l'apport de CAPTEN par rapport à une approche par workflow classique**

Une seule réponse possible.

- ◯ Ne sais pas quantifier l'apport de CAPTEN sur une approche par workflow

113. **Pensez-vous que CAPTEN peut remplacer ces approches ?** *

Une seule réponse possible.

- ◯ Oui
- ◯ Non

114. **Si non, pourquoi ?**

_____

_____

_____

_____

_____

115. **Commentaire général**

_____

_____

_____

_____

_____

F Compléments concernant la recherche avancée CAPTEN-FRUIT

F.1 Patrons de recherche

F.1.1 Desiderata

Exemple 1.1 (Patron de recherche des *desiderata*)

$$\mathcal{P}_{\mathcal{D}_d} = \langle (Connaissance, 0.9), (Objectif, 0.8), (Analyse, 0.9), (Addendum, 0.1), (Nom, 0.2), (Graphe\_de\_variables, 0.4), (Patron\_de\_sortie, 0.3) \rangle \quad \text{(F.1)}$$

F.1.2 Contexte

Exemple 1.2 (Patron de recherche du contexte)

$$\mathcal{P}_{\mathcal{D}_d} = \langle (\text{Éléments\_Narratifs\_Contextuels}, 0.9), (Graphe\_de\_Variables\_d'entrée, 0.7), (Patron\_d'entrée, 0.3)(Nom, 0.2)(Addendum, 0.1) \rangle \quad \text{(F.2)}$$

F.2 Requêtes SPARQL

Dans cette section, nous présentons les requêtes utilisées dans notre prototype et qui sont actionnées par les patrons de recherche pour réaliser les recherches correspondantes. Il s'agit ici des clauses *WHERE* de ces requêtes exprimées dans le code source de notre prototype ; nous récupérons toutes les variables (i.e. les termes préfixés par *?*, comme *?obj*) : c'est la raison pour laquelle nous n'indiquons pas le sélecteur à chaque fois. De plus, ces requêtes possèdent principalement deux variantes : *termReady* pour interroger directement les éléments en lien avec l'entité concernée, et la variante *relationReady* pour explorer les éléments narratifs qui lui sont associés.

Nous présentons aussi dans la dernière sous-Section F.2.10 la fonction destinée à récupérer l'analyse parente d'un élément. Par exemple, si un opérateur narré est identifié comme résultat, son élément parent est une étape, qui elle-même possède un processus d'analyse narré comme parent. Cette information est capitale pour établir le backtrack afin d'assister l'utilisateur et permettre l'explication des résultats [1].

F.2.1 Nom

```
nameDeferencing.prototype.relationReady = function()
{
  return "?s <http://www.CAPTEN.org/SEED/ontologies/hasName> ?obj . ?obj <http://
      ↪ www.w3.org/1999/02/22-rdf-syntax-ns#li> ?x . ?obj <http://www.w3.org
      ↪ /1999/02/22-rdf-syntax-ns#li> ?y .";
}
nameDeferencing.prototype.termReady = function()
```

1. Pour rappel, l'ensemble du code source est disponible en ligne (Lebis, 2018[e])

```
{
  return "?s <http://www.CAPTEN.org/SEED/ontologies/hasName> ?obj . ?obj <http://
      ↪ www.w3.org/1999/02/22-rdf-syntax-ns#li> ?x .";
}
```

F.2.2 Graphe de variables

```
rgteDeferencing.prototype.relationReady = function()
{
  return "?g <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.CAPTEN
      ↪ .org/SEED/ontologies/RGTE> . ?g  <http://www.CAPTEN.org/SEED/ontologies/
      ↪ hasVariable> ?x . ?g  <http://www.CAPTEN.org/SEED/ontologies/hasVariable
      ↪ > ?y .";
}

rgteDeferencing.prototype.termReady = function()
{
    return "?g <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.
        ↪ CAPTEN.org/SEED/ontologies/RGTE>. ?g <http://www.CAPTEN.org/SEED/
        ↪ ontologies/hasVariable> ?x .";
}
```

F.2.3 Graphe de variables d'entrée

```
inputGraphDeferencing.prototype.relationReady = function()
{
  return "?e <http://www.CAPTEN.org/SEED/ontologies/hasInput> ?g . ?g  <http://
      ↪ www.CAPTEN.org/SEED/ontologies/hasVariable> ?x . ?g  <http://www.CAPTEN.
      ↪ org/SEED/ontologies/hasVariable> ?y .";
}

inputGraphDeferencing.prototype.termReady = function()
{
    return "?e <http://www.CAPTEN.org/SEED/ontologies/hasInput> ?g . ?g <http://
        ↪ www.CAPTEN.org/SEED/ontologies/hasVariable> ?x .";
}
```

F.2.4 Connaissance

```
knowledgeDeferencing.prototype.relationReady = function()
{
  return "?g <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.CAPTEN
      ↪ .org/SEED/ontologies/Knowledge> . ?g  <http://www.CAPTEN.org/SEED/
      ↪ ontologies/hasVariable> ?x . ?g  <http://www.CAPTEN.org/SEED/ontologies/
      ↪ hasVariable> ?y .";
}

knowledgeDeferencing.prototype.termReady = function()
{
    //quasi equivalent to: generatedKnowledgeDeferencing.prototype.termReady.
        ↪ give a graph in addition
    return "?a <http://www.CAPTEN.org/SEED/ontologies/custom/knowledgeGeneratedBy
        ↪ > ?x . ?g <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://
        ↪ www.CAPTEN.org/SEED/ontologies/Knowledge> . ?g <http://www.CAPTEN.org/
        ↪ SEED/ontologies/hasVariable> ?x . ";
}

generatedKnowledgeDeferencing.prototype.termReady = function()
```

```
{
  return "?a <http://www.CAPTEN.org/SEED/ontologies/custom/knowledgeGeneratedBy>
      ↪ ?x .";
}
```

F.2.5 Éléments contextuels narratifs

```
contextDeferencing.prototype.relationReady = function()
{
  return "?s <http://www.CAPTEN.org/SEED/ontologies/hasContext> ?c . ?c <http://
      ↪ www.CAPTEN.org/SEED/ontologies/hasContent> ?content . ?content <http://
      ↪ www.w3.org/1999/02/22-rdf-syntax-ns#li> ?x . ?content <http://www.w3.org
      ↪ /1999/02/22-rdf-syntax-ns#li> ?y .";
}

contextDeferencing.prototype.termReady = function()
{
  return "?s <http://www.CAPTEN.org/SEED/ontologies/hasContext> ?c . ?c <http://
      ↪ www.CAPTEN.org/SEED/ontologies/hasContent> ?content . ?content <http://
      ↪ www.w3.org/1999/02/22-rdf-syntax-ns#li> ?x .";
}
```

F.2.6 Objectif

```
objectiveDeferencing.prototype.relationReady = function()
{
  return "?s <http://www.CAPTEN.org/SEED/ontologies/hasObjective> ?obj . ?obj <
      ↪ http://www.CAPTEN.org/SEED/ontologies/hasContent> ?content . ?content <
      ↪ http://www.w3.org/1999/02/22-rdf-syntax-ns#li> ?x . ?content <http://www
      ↪ .w3.org/1999/02/22-rdf-syntax-ns#li> ?y .";
}

objectiveDeferencing.prototype.termReady = function()
{
  return "?s <http://www.CAPTEN.org/SEED/ontologies/hasObjective> ?obj . ?obj <
      ↪ http://www.CAPTEN.org/SEED/ontologies/hasContent> ?content . ?content <
      ↪ http://www.w3.org/1999/02/22-rdf-syntax-ns#li> ?x .";
}
```

F.2.7 Addendum

```
addendumDeferencing.prototype.relationReady = function()
{
  return "?s <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.CAPTEN
      ↪ .org/SEED/ontologies/Addendum> . ?s <http://www.CAPTEN.org/SEED/
      ↪ ontologies/hasContent> ?content . ?content <http://www.w3.org
      ↪ /1999/02/22-rdf-syntax-ns#li> ?x . ?content <http://www.w3.org
      ↪ /1999/02/22-rdf-syntax-ns#li> ?y .";
}

addendumDeferencing.prototype.termReady = function()
{
  return "?s <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.CAPTEN
      ↪ .org/SEED/ontologies/Addendum> . ?s <http://www.CAPTEN.org/SEED/
      ↪ ontologies/hasContent> ?content . ?content <http://www.w3.org
      ↪ /1999/02/22-rdf-syntax-ns#li> ?x .";
}
```

F.2.8 Patron de sortie

```
outputBehaviourDeferencing.prototype.relationReady = function()
{
  return "?o <http://www.CAPTEN.org/SEED/ontologies/hasOutputBehaviour> ?g . ?g <
      ↪ http://www.CAPTEN.org/SEED/ontologies/hasVariable> ?x . ?g <http://www.
      ↪ CAPTEN.org/SEED/ontologies/hasVariable> ?y ."
}

outputBehaviourDeferencing.prototype.termReady = function()
{
  return "?o <http://www.CAPTEN.org/SEED/ontologies/hasOutputBehaviour> ?g . ?g <
      ↪ http://www.CAPTEN.org/SEED/ontologies/hasVariable> ?x .";
}
```

F.2.9 Patron d'entrée

```
inputBehaviourDeferencing.prototype.relationReady = function()
{
  return "?o <http://www.CAPTEN.org/SEED/ontologies/hasInputBehaviour> ?g . ?g <
      ↪ http://www.CAPTEN.org/SEED/ontologies/hasVariable> ?x . ?g <http://www.
      ↪ CAPTEN.org/SEED/ontologies/hasVariable> ?y ."
}

inputBehaviourDeferencing.prototype.termReady = function()
{
  return "?o <http://www.CAPTEN.org/SEED/ontologies/hasInputBehaviour> ?g . ?g <
      ↪ http://www.CAPTEN.org/SEED/ontologies/hasVariable> ?x .";
}
```

F.2.10 Analyses parentes

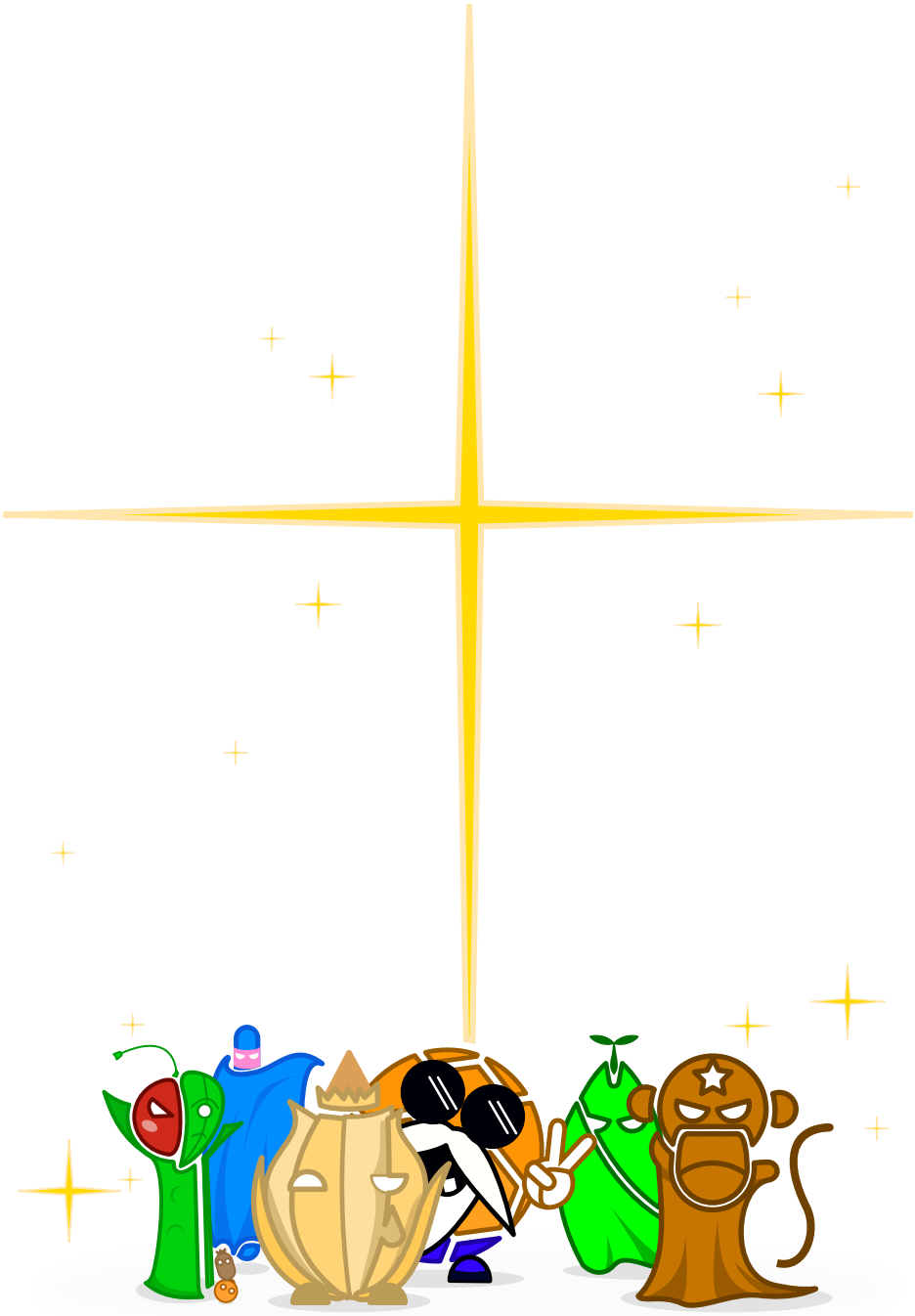
```
stepDeferencing.prototype.getParentAnalyses = async function(IDElmt)
{
  return new Promise(
    async function(resolve, reset)
    {
      if(!IDElmt)
      {
        resolve([]);
        return;
      }

      var query = "SELECT * WHERE { ?a <http://www.w3.org/1999/02/22-rdf-syntax-
          ↪ ns#type> <http://www.CAPTEN.org/SEED/ontologies/
          ↪ NarratedAnalysisProcess> . ?a <http://www.w3.org/1999/02/22-rdf-
          ↪ syntax-ns#li> <"+IDElmt+"> .}";
      var res = await HYLAR_HANDLER.promiseToQueryOnto(query);
      var parent = [];

      for(var i in res)
      {
        parent.push(res[i].a.value);
      }

      resolve(parent);
    }
  );
}
```

Colophon

This thesis has been funded by the ANR as ANR-14-CE24-0015 (Projet HUBBLE).

This thesis was typeset with LaTeX 2$_\varepsilon$. It uses the *Clean Thesis* style developed by Ricardo Langner. The design of the *Clean Thesis* style is inspired by user guide documents from Apple Inc.

Download the *Clean Thesis* style at http://cleanthesis.der-ric.de/.



Paris, Le 22 Mai 2019

Alexis Lebis