Corpus de traces d'activité dans les environnements informatiques pour l'apprentissage humain : modélisation, étude d'une plateforme de gestion, et application à la construction de corpus de référence.

Hajer Chebil

Ecole Nationale Supérieure des Mines
SAINT-ETIENNE

NNT : *2013 EMSE 0718*

THÈSE

présentée par

Hajer CHEBIL

pour obtenir le grade de
Docteur de l'École Nationale Supérieure des Mines de Saint-Étienne

Spécialité : Informatique

CORPUS DE TRACES D'ACTIVITÉ DANS LES ENVIRONNEMENTS INFORMATIQUES POUR L'APPRENTISSAGE HUMAIN : MODÉLISATION ÉTUDE D'UNE PLATEFORME DE GESTION ET APPLICATION À LA CONSTRUCTION DE CORPUS DE RÉFÉRENCE

soutenue à Saint-Étienne, le 29 octobre 2013

Membres du jury

Président :	Philippe VIDAL	Professeur des Universités, Université Paul SABATIER – Toulouse 3, Toulouse
Rapporteurs :	Alain MILLE	Professeur des Universités, Université Claude Bernard Lyon 1, Lyon
	Vanda LUENGO	Maître de Conférences HDR, Université Joseph Fourier, Grenoble
Examinateur(s) :	Philippe VIDAL	Professeur des Universités, Université Paul SABATIER – Toulouse 3, Toulouse
Directeur(s) de thèse :	Jean-Jacques GIRARDOT	Directeur de Recherche 2, École Nationale Supérieure des Mines de Saint-Étienne
	Christophe COURTIN	Maître de Conférences, Université de Savoie, Chambéry
Invité(s) :	Christophe REFFAY	Maître de Conférences, Université de Franche-Comté, Besançon

Spécialités doctorales :	Responsables :
SCIENCES ET GENIE DES MATERIAUX	K. Wolski Directeur de recherche
MECANIQUE ET INGENIERIE	S. Drapier, professeur
GENIE DES PROCEDES	F. Gruy, Maître de recherche
SCIENCES DE LA TERRE	B. Guy, Directeur de recherche
SCIENCES ET GENIE DE L'ENVIRONNEMENT	D. Graillot, Directeur de recherche
MATHEMATIQUES APPLIQUEES	O. Roustant, Maître-assistant
INFORMATIQUE	O. Boissier, Professeur
IMAGE, VISION, SIGNAL	JC. Pinoli, Professeur
GENIE INDUSTRIEL	A. Dolgui, Professeur
MICROELECTRONIQUE	

EMSE : Enseignants-chercheurs et chercheurs autorisés à diriger des thèses de doctorat (titulaires d'un doctorat d'État ou d'une HDR)

AVRIL	Stéphane	PR2	Mécanique et ingénierie	CIS
BATTON-HUBERT	Mireille	PR2	Sciences et génie de l'environnement	FAYOL
BENABEN	Patrick	PR1	Sciences et génie des matériaux	CMP
BERNACHE-ASSOLLANT	Didier	PR0	Génie des Procédés	CIS
BIGOT	Jean Pierre	MR(DR2)	Génie des Procédés	SPIN
BILAL	Essaid	DR	Sciences de la Terre	SPIN
BOISSIER	Olivier	PR1	Informatique	FAYOL
BORBELY	Andras	MR(DR2)	Sciences et génie de l'environnement	SMS
BOUCHER	Xavier	PR2	Génie Industriel	FAYOL
BRODHAG	Christian	DR	Sciences et génie de l'environnement	FAYOL
BURLAT	Patrick	PR2	Génie Industriel	FAYOL
COLLOT	Philippe	PR0	Microélectronique	CMP
COURNIL	Michel	PR0	Génie des Procédés	DIR
DARRIEULAT	Michel	IGM	Sciences et génie des matériaux	SMS
DAUZERE-PERES	Stéphane	PR1	Génie Industriel	CMP
DEBAYLE	Johan	CR	Image Vision Signal	CIS
DELAFOSSE	David	PR1	Sciences et génie des matériaux	SMS
DESRAYAUD	Christophe	PR2	Mécanique et ingénierie	SMS
DOLGUI	Alexandre	PR0	Génie Industriel	FAYOL
DRAPIER	Sylvain	PR1	Mécanique et ingénierie	SMS
FEILLET	Dominique	PR2	Génie Industriel	CMP
FOREST	Bernard	PR1	Sciences et génie des matériaux	CIS
FORMISYN	Pascal	PR0	Sciences et génie de l'environnement	DIR
FRACZKIEWICZ	Anna	DR	Sciences et génie des matériaux	SMS
GARCIA	Daniel	MR(DR2)	Génie des Procédés	SPIN
GERINGER	Jean	MA(MDC)	Sciences et génie des matériaux	CIS
GIRARDOT	Jean-jacques	MR(DR2)	Informatique	FAYOL
GOEURIOT	Dominique	DR	Sciences et génie des matériaux	SMS
GRAILLOT	Didier	DR	Sciences et génie de l'environnement	SPIN
GROSSEAU	Philippe	DR	Génie des Procédés	SPIN
GRUY	Frédéric	PR1	Génie des Procédés	SPIN
GUY	Bernard	DR	Sciences de la Terre	SPIN
GUYONNET	René	DR	Génie des Procédés	SPIN
HAN	Woo-Suck	CR	Mécanique et ingénierie	SMS
HERRI	Jean Michel	PR1	Génie des Procédés	SPIN
INAL	Karim	PR2	Microélectronique	CMP
KERMOUCHE	Guillaume	PR2	Mécanique et Ingénierie	SMS
KLOCKER	Helmut	DR	Sciences et génie des matériaux	SMS
LAFOREST	Valérie	MR(DR2)	Sciences et génie de l'environnement	FAYOL
LERICHE	Rodolphe	CR	Mécanique et ingénierie	FAYOL
LI	Jean Michel		Microélectronique	CMP
MALLIARAS	Georges	PR1	Microélectronique	CMP
MOLIMARD	Jérôme	PR2	Mécanique et ingénierie	CIS
MONTHEILLET	Franck	DR	Sciences et génie des matériaux	SMS
PERIER-CAMBY	Laurent	PR2	Génie des Procédés	DFG
PIJOLAT	Christophe	PR0	Génie des Procédés	SPIN
PIJOLAT	Michèle	PR1	Génie des Procédés	SPIN
PINOLI	Jean Charles	PR0	Image Vision Signal	CIS
POURCHEZ	Jérémy	CR	Génie des Procédés	CIS
ROUSTANT	Olivier	MA(MDC)		FAYOL
STOLARZ	Jacques	CR	Sciences et génie des matériaux	SMS
SZAFNICKI	Konrad	MR(DR2)	Sciences et génie de l'environnement	CMP
TRIA	Assia		Microélectronique	CMP
VALDIVIESO	François	MA(MDC)	Sciences et génie des matériaux	SMS
VIRICELLE	Jean Paul	MR(DR2)	Génie des Procédés	SPIN
WOLSKI	Krzystof	DR	Sciences et génie des matériaux	SMS
XIE	Xiaolan	PR1	Informatique	CIS

ENISE : Enseignants-chercheurs et chercheurs autorisés à diriger des thèses de doctorat (titulaires d'un doctorat d'État ou d'une HDR)

BERGHEAU	Jean-Michel	PU	Mécanique et Ingénierie	ENISE
BERTRAND	Philippe	MCF	Génie des procédés	ENISE
DUBUJET	Philippe	PU	Mécanique et Ingénierie	ENISE
FORTUNIER	Roland	PR	Sciences et Génie des matériaux	ENISE
GUSSAROV Andrey	Andrey	Enseignant contractuel	Génie des procédés	ENISE
HAMDI	Hédi	MCF	Mécanique et Ingénierie	ENISE
LYONNET	Patrick	PU	Mécanique et Ingénierie	ENISE
RECH	Joël	MCF	Mécanique et Ingénierie	ENISE
SMUROV	Igor	PU	Mécanique et Ingénierie	ENISE
TOSCANO	Rosario	MCF	Mécanique et Ingénierie	ENISE
ZAHOUANI	Hassan	PU	Mécanique et Ingénierie	ENISE

Mise à jour : 07/01/2013

PR 0	Professeur classe exceptionnelle
PR 1	Professeur 1ère classe
PR 2	Professeur 2ème classe
PU	Professeur des Universités
MA (MDC)	Maître assistant
DR	Directeur de recherche

Ing.	Ingénieur
MCF	Maître de conférences
MR (DR2)	Maître de recherche
CR	Chargé de recherche
EC	Enseignant-chercheur
IGM	Ingénieur général des mines

SMS	Sciences des Matériaux et des Structures
SPIN	Sciences des Processus Industriels et Naturels
FAYOL	Institut Henri Fayol
CMP	Centre de Microélectronique de Provence
CIS	Centre Ingénierie et Santé

Remerciements

Je remercie les membres du jury d'avoir accepté d'évaluer ce travail. Merci à Philippe Vidal d'avoir accepté de présider le jury. Merci à mes deux rapporteurs Vanda Luengo et Alain Mille pour leurs conseils et remarques constructives. Merci à Christophe Reffay pour sa lecture et ses remarques très constructives.

Je remercie l'École des Mines de Saint-Étienne pour m'avoir donné l'occasion de m'inscrire en thèse, et l'Université de Savoie de m'avoir accueillie. Merci à tous les membres de l'équipe Syscom pour son accueil et son aide.

Un grand merci à mes directeurs de thèse Jean-Jacques Girardot et Christophe Courtin pour leur encadrement, disponibilité et conseils. Je ne les remercierai jamais assez pour leur grand soutien sur les plans professionnels et personnels.

Je remercie ma famille et mes amies de m'avoir supportée pendant cette période qui a été parfois difficile. Merci à mes parents et ma sœur pour leur amour et pour avoir toujours cru en moi et n'ont cessé de m'encourager.

Enfin, je remercie mon mari Hichem pour sa patience, son sacrifice, son soutien, sa présence, et son amour. C'est en grande partie grâce à lui que j'ai pu aller jusqu'au bout de ce travail.

Table des matières

Table des illustrations

Première partie
Problématique et État de l’art

Chapitre 1 : Problématique

1.1 Introduction

Dans ce premier chapitre, nous présentons le contexte de ce travail de thèse ainsi que la problématique étudiée. Nous exposons ensuite les trois questions de recherche étudiées dans l'objectif de répondre à cette problématique. Enfin, nous explicitons les motivations qui sont à l'origine de ce travail de recherche.

1.2 Contexte

Le travail présenté dans cette thèse est réalisé dans le cadre du projet « Personnalisation des environnements informatiques pour l'apprentissage humain (EIAH) »[1] faisant partie du cluster « Informatique, Signal, Logiciel Embarqué »[2] financé par la région Rhône-Alpes. Ce

[1] http://cluster-isle-eiah.liris.cnrs.fr/
[2] http://cluster-isle.grenoble-inp.fr/

projet s'intéresse à l'importance de la personnalisation dans le développement des EIAH, en se basant sur les traces d'interaction considérées comme source très importante permettant de renseigner sur le déroulement de l'activité d'apprentissage.

Ce projet comprend cinq tâches fortement liées ayant comme objectif commun la contribution à la personnalisation des EIAH en utilisant les traces d'interaction d'apprentissage dans l'étude de problématiques de recherche différentes. La première tâche considère la trace comme objet d'étude et propose un cadre théorique permettant la modélisation, la transformation et la gestion de la trace modélisée (Settouti, 2011) (Settouti et al., 2009). La deuxième tâche s'intéresse au couplage entre les traces d'interaction collectées et les scénarios pédagogiques définis lors de la conception d'une situation d'apprentissage. Un scénario pédagogique définit la tâche prescrite par l'enseignant concepteur alors que la trace collectée représente la tâche effective réalisée par les apprenants. Les traces permettent donc de vérifier l'adéquation entre le scénario prescrit et le scénario effectif (Ferraris et Lejeune, 2009) pouvant ainsi aider un enseignant à réguler la conception du scénario. La troisième tâche s'intéresse au diagnostic des connaissances à partir des traces d'interaction. Le diagnostic permet d'identifier les connaissances mobilisées d'un domaine étudié dans le but de superviser l'activité d'apprentissage. Le diagnostic des connaissances permet l'adaptation de l'environnement d'apprentissage en le personnalisant en fonction des profils cognitifs calculés à partir des traces (Settouti et al., 2011) (Lalle et al., 2013). La quatrième tâche utilise les traces d'interaction pour analyser et améliorer le déroulement d'une interaction médiatisée par ordinateur entre les tuteurs et apprenants à distance. Les interactions analysées sont collectées à partir d'outils de communication utilisés dans un contexte d'apprentissage (May et al., 2008).

Notre travail fait partie de la cinquième tâche qui se place dans la continuité des tâches précédentes et a comme objectif la proposition d'un modèle générique de partage de corpus de traces d'interaction prenant en compte l'hétérogénéité de ces traces due aux différents types d'environnements d'apprentissage et à la variété des domaines d'application étudiés, ainsi que les besoins de contextualisation liés au partage. L'objectif de cette tâche étant de faire converger les résultats des différentes tâches et de s'ouvrir sur la communauté des chercheurs menant leurs recherches à l'aide des EIAH, la solution proposée doit également permettre aux chercheurs d'utiliser des outils d'analyse développés par d'autres chercheurs pour réaliser une nouvelle analyse, la vérification de résultats d'analyse publiés, des analyses complémentaires, ou comparer différentes méthodes d'analyse. Le modèle proposé doit servir

de base pour la proposition d'une architecture de plate-forme, prenant en compte les problèmes d'interopérabilité, qui permette le partage entre chercheurs de corpus de traces d'interaction d'apprentissage et d'outils d'analyse de ces corpus, ainsi que les analyses réalisées sur ces corpus.

1.3 Problématique

Notre travail s'intéresse principalement au partage de traces d'interaction et d'outils d'analyse entre chercheurs utilisant dans leurs travaux de recherche des environnements informatiques pour l'apprentissage humain. Les chercheurs utilisant les EIAH tirent parti du caractère technique de ces environnements pour collecter des traces d'interaction relatives à l'activité des acteurs participant à une expérimentation d'apprentissage. Ces traces d'interaction représentent une source d'information importante sur le déroulement de l'apprentissage, et sont indispensables pour l'observation de l'activité d'apprentissage utilisant un EIAH. La collecte des traces d'interaction produites par un EIAH se fait généralement d'une manière ad-hoc pour répondre à des besoins de recherche et d'observation spécifiques et parfois très liés au domaine d'application. En effet, les traces collectées ont des modèles différents suivant leur contenu sémantique et peuvent être structurées suivant différents formats. Par ailleurs, les chercheurs développent généralement des outils d'analyse fortement couplés aux environnements d'apprentissage (Martínez et al., 2005) pour répondre à des besoins d'analyse spécifiques. Cela rend le partage, la réutilisation et la comparaison des résultats de différentes équipes de recherche difficiles.

Le travail présenté dans cette thèse s'intéresse donc aux problématiques de (1) partage de corpus de traces d'interaction d'apprentissage, (2) partage d'outils d'analyse et étude de leur interopérabilité sur les corpus partagés, et (3) liaison entre les corpus partagés et les travaux d'analyse réalisés sur ces corpus.

1.4 Questions de recherche

Afin de répondre à la problématique présentée, nous avons identifié trois questions de recherche nous permettant d'étudier les différentes facettes du problème.

1.4.1 Modèle d'un corpus

La première question de recherche s'intéresse à la modélisation d'un corpus de traces d'interaction d'apprentissage. Ce travail pose naturellement la question de la représentation et de la structure des données à partager. En effet, il est nécessaire de préciser ce que peut contenir un corpus partagé, ainsi que les métadonnées nécessaires à sa description et celle de ses composants. Un corpus peut être défini comme une collection composée de ressources contenant des traces, des ressources contextuelles relatives à une expérimentation d'apprentissage, et éventuellement des ressources relatives à des analyses réalisées sur le corpus. La structuration d'un corpus de données d'apprentissage représente la problématique principale étudiée dans le cadre du projet MULCE (Reffay et al. 2008) dont nous nous sommes inspirés. La première question de recherche peut donc être formulée ainsi : « Comment modéliser un corpus de traces d'interaction d'apprentissage en vue de son partage et de son analyse, en prenant en considération l'importance de la contextualisation des données partagées ? ».

1.4.2 Interopérabilité entre corpus et outils d'analyse partagés

La deuxième question de recherche traite du problème d'interopérabilité entre les corpus partagés et les outils d'analyse provenant de différentes équipes de recherche, outils qui n'ont a priori pas été conçus pour être interopérables. En effet, il est souvent intéressant qu'un chercheur puisse réutiliser, sur ses propres données, un outil d'analyse développé par une autre équipe de recherche. Cela lui permettrait de profiter des fonctionnalités offertes par un nouvel outil d'analyse, ou de comparer les résultats d'analyse produits par différents outils. Nous désignons essentiellement par outil d'analyse une application informatique utilisée par un chercheur pour analyser des traces d'apprentissage. Cependant, dans certaines disciplines des sciences humaines, un chercheur peut utiliser des méthodes d'analyse non implémentées et dont la réutilisation par d'autres chercheurs peut s'avérer utile. Nous considérons donc qu'un outil d'analyse peut aussi correspondre à une méthode d'analyse non implémentée dont la sémantique des données en entrée et en sortie est définie. Il convient alors de répondre à la question : « Comment garantir une interopérabilité entre les corpus partagés et les outils d'analyse sans imposer une nouvelle représentation des traces d'interaction comme condition à la réalisation du partage ? ».

1.4.3 Intégration et partage des travaux d'analyse

La troisième question concerne les travaux d'analyse réalisés sur les corpus partagés et l'intérêt de partager les données correspondantes, de les lier et de les intégrer aux corpus concernés. Suivant les besoins des chercheurs, une analyse peut impliquer un ou plusieurs outils d'analyse. Ce partage permet aux chercheurs de vérifier des résultats publiés ou de réaliser des analyses cumulatives ou comparatives. La question étudiée est : « Comment représenter et intégrer les travaux d'analyse réalisés sur un ou plusieurs corpus de manière qu'ils soient réutilisables par d'autres chercheurs pour vérification, comparaison, ou analyse complémentaire ? ».

1.5 Motivations

Les trois questions de recherche identifiées sont liées à trois motivations principales justifiant le besoin de partage de corpus de traces d'interaction et d'outils d'analyse, que nous décrivons ci-dessous.

1.5.1 Accès à des corpus de traces d'interaction d'apprentissage contextualisées partagés entre équipes de différentes disciplines

Il est important de pouvoir partager les données expérimentales entre chercheurs utilisant des EIAH dans leurs recherches qui peuvent concerner différentes disciplines (didactique, psychologie, informatique, etc.). En effet, le montage d'une expérimentation en situation écologique (c'est-à-dire réalisée en situation réelle et authentique d'apprentissage et non expérimentale dans un laboratoire) étant compliqué et très coûteux en termes de temps (nombre d'acteurs impliqués, durée de l'expérimentation, dispositif technique, etc.), le partage de données issues d'expérimentations passées intéresse probablement des chercheurs d'une même communauté ou d'une communauté voisine (Reffay et al., 2008). La disponibilité des traces et données issues d'expérimentation permet à la communauté scientifique d'accéder aux données sources pour reproduire des résultats d'analyses publiés, ce qui offre la possibilité de les confirmer, les contredire et surtout les enrichir et les capitaliser (projet dataTel (Drachsler et al., 2010)) (Projet Datashop, Koedinger et al., 2008) (Corbel et al., 2006). Par ailleurs, des chercheurs travaillant sur des outils d'analyse, ou la comparaison d'outils d'analyse, ont souvent besoin de grandes quantités de données pour valider leurs

modèles et comparer différentes méthodes d'analyse (p. ex. benchmark pour la comparaison de différentes techniques de diagnostic cognitif dans les systèmes de type tuteurs intelligents ITS[3] (Lalle et al., 2013)). Des données d'interaction représentatives partagées peuvent aussi être utiles pour une étape de calibrage de dispositifs d'apprentissage utilisant des outils métacognitifs ou de guidage (Reffay et Betbeder, 2009), en fixant des seuils utiles dans l'évaluation de l'activité.

1.5.2 Réutilisation et comparaison d'outils d'analyse

Les outils d'analyse sont souvent fortement couplés aux environnements d'apprentissage (Projets du réseau d'excellence Kaleidoscope IA JEIRP (Martínez et al., 2005), CAVICOLA (Harrer et al., 2007)) ce qui pose un problème de réutilisation entre différentes équipes de recherche. Les outils d'analyse, développés pour répondre à des besoins spécifiques, restent souvent à un niveau prototypique (Reffay et Betbeder, 2009) ce qui empêche leur évolution et une utilisation élargie. Il est donc intéressant de permettre aux chercheurs de réutiliser des outils d'analyse pertinents pour leurs études, et de résoudre, au niveau de la gestion des corpus, les problèmes d'interopérabilité afférents.

[3] Intelligent Tutoring Systems

1.5.3 Accès aux analyses réalisées sur les corpus pour vérification, comparaison et capitalisation

Les analyses réalisées sur des corpus de traces produits par des expérimentations font souvent l'objet de publications au sein de la communauté scientifique. Le partage des données relatives aux analyses rend possible l'accès aux détails relatifs à leur déroulement (descriptions, données utilisées et données produites). Cela permet aux chercheurs de vérifier l'exactitude des résultats, de les comparer à d'autres résultats, ou de les enrichir par des analyses cumulatives.

1.6 Conclusion

Dans ce travail de thèse, nous apportons des solutions pour répondre aux trois questions de recherche présentées ci-dessus. Nous commençons par étudier les travaux existants qui se sont intéressés à la modélisation des traces issues des environnements informatiques pour l'apprentissage humain, et au partage de corpus de traces et d'outils d'analyse. En nous basant sur les conclusions de cette étude, nous proposons une modélisation de corpus et une approche nouvelle baptisée approche par « proxy » (Chebil et al. 2012a) (Chebil et al. 2012b) permettant le partage de corpus de traces d'apprentissage contextualisées, et leur analyse avec des outils partagés. En nous basant sur cette approche, nous proposons l'architecture de la plateforme « Beatcorp » (Benchmarking platform for Analysis of Traces Corpora) de partage de corpus et d'outils d'analyse.

Chapitre 2 : État de l'art

2.1 Introduction

Ce chapitre est dédié à une étude de différents travaux liés à notre problématique de recherche. Cet état de l'art est composé de deux parties principales. Dans la première partie, nous présentons différents travaux s'intéressant à la modélisation de la trace dans le domaine des EIAH. En effet, la première question que l'on se pose quand on veut partager des données est la représentation qu'auront ces données. Parmi ces travaux, certains proposent des modèles génériques permettant de représenter des traces provenant de n'importe quel outil, alors que d'autres proposent des modèles spécifiques liés au type d'outil et/ou au domaine d'application. La deuxième partie de cette étude concerne les projets qui se sont intéressés à la problématique du partage dans la communauté EIAH de corpus de traces d'interaction et d'outils d'analyse de ces corpus. Enfin nous clôturons ce chapitre par des conclusions sur l'état de l'art.

2.2 Traces et modèles de traces

Les traces sont souvent collectées par un module de collecte interne à l'EIAH, et possèdent la plupart du temps des formats propriétaires très liés au fonctionnement de l'EIAH, sans soucis de réutilisation. Par ailleurs, les traces peuvent être analysées par des outils d'analyse et de visualisation externes à l'EIAH. Mais ces outils restent généralement fortement couplés aux EIAH ayant généré les traces. Ce couplage fort entre EIAH et outils d'analyses et visualisations est dû à l'absence d'un modèle et de formats standards de représentation des traces d'interaction issues d'un EIAH. L'absence d'un tel modèle s'explique par le fait que la majorité des systèmes utilisent les traces comme moyen d'observation et non comme objectif d'étude, donc chaque concepteur définit le format relatif au modèle qui lui convient selon ses besoins. Une autre raison est justement liée à la diversité des besoins d'observation, ce qui rend très difficile de se mettre d'accord sur un modèle standard qui soit (1) utilisable par un grand nombre de systèmes de traçage d'une part et d'analyse et de visualisation de l'autre, et (2) pertinent pour les analyses spécifiques propres à chaque EIAH et aux questions de recherche afférentes.

Dans le domaine des EIAH, certains travaux se sont intéressés à l'étude des traces d'interaction issues de situations d'apprentissage et leur représentation par la proposition de modèles plus ou moins génériques que nous détaillerons dans ce qui suit.

2.2.1 MUSETTE

MUSETTE (*Modeling USEs and Tasks for Tracing Experience* ou Modélisation des usages et tâches pour tracer l'expérience) (Champin et al., 2004) est une approche de modélisation et de réutilisation de l'expérience d'utilisation d'un système informatique à partir de la trace. Cette approche a été proposée dans le cadre du développement d'assistants logiciels et considère la trace d'utilisation comme une source de connaissance qui enregistre l'expérience passée. Les traces d'utilisation sont considérées comme une base de connaissance pouvant être exploitée en utilisant le paradigme du raisonnement à partir de cas pour réutiliser l'expérience. Une trace correspond à une séquence alternée d'états et de transitions. Un état est composé d'un ensemble d'objets utilisés par l'utilisateur dans son interaction avec le système informatique. Une transition est quant à elle composée d'un ensemble d'événements. Les objets et événements peuvent être liés par des relations.

L'approche (cf. Figure 1) se base sur trois notions principales, à savoir le modèle d'utilisation, le modèle d'observation, et la signature de tâche expliquée. Le modèle d'utilisation définit les éléments pouvant constituer une trace, ces éléments peuvent être des objets, des événements, ou des relations. Le modèle d'observation définit la manière dont la trace est effectivement construite. La signature de tâche expliquée permet d'identifier une partie de la trace relative à une tâche particulière appelée épisode et qui peut être réutilisée dans des situations similaires. L'agent observateur observe le changement de l'état du système conformément à un modèle d'observation. Il génère une trace primitive conforme à un modèle d'utilisation général. Ensuite, un analyseur de trace générique extrait des épisodes signifiants à partir de la trace primitive conformément aux signatures de tâches expliquées. Ces épisodes peuvent être réutilisés par des agents assistants.

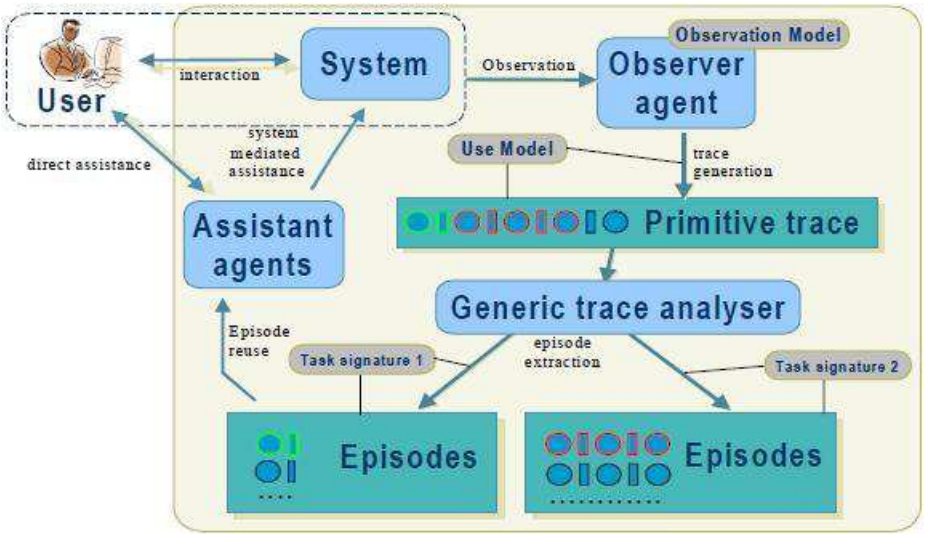


Figure 1 Approche MUSETTE (Champin et al., 2004)

Cette approche s'intéresse à la modélisation de la trace et à sa réutilisation comme expérience passée par les agents assistants. L'originalité de ce travail est la généralisation du traitement de la trace en le considérant comme un processus indépendant de l'environnement d'apprentissage. Toutefois, les agents observateurs, ainsi que les modèles d'observation sont définis de manière ad-hoc et sont donc fortement lié aux environnements d'apprentissage utilisés. Par ailleurs, ce travail propose une approche générique qui donne un cadre conceptuel pour la modélisation de la trace d'utilisation qui doit être adaptée aux différentes applications.

2.2.2 Systèmes à Base de Traces Modélisées[4]

Ce travail constitue en partie une suite au travail sur l'approche MUSETTE. Dans (Settouti et al., 2009), une trace d'interaction est définie comme une séquence d'éléments observés enregistrés suite à l'interaction et la navigation d'un utilisateur dans un EIAH. Un observé correspond à une donnée structurée issue de l'observation. Les auteurs avaient déjà distingué dans (Settouti et al., 2007) deux types de traces : « l'histoire interactionnelle d'un apprenant utilisant un EIAH » ou « les productions qu'il a laissées lors de son apprentissage ». Les auteurs se placent dans le premier cas, car ils s'intéressent à la propriété descriptive de la trace, illustrée à partir de sa capacité à retracer l'interaction, plutôt que sa propriété résiduelle qui considère la trace comme un résultat de l'activité d'apprentissage.

[4] Ce travail a été réalisé dans le cadre de la tâche 1 « Production des traces et représentation » du projet ISLE/EIAH de la région Rhône-Alpes.

La collecte des traces peut offrir différents services selon son destinataire ; ainsi, dans (Settouti et al., 2007), quatre utilisations principales des traces d'interaction ont été identifiées : (1) prendre conscience (*awareness*) de l'activité par l'apprenant et l'enseignant (Scheffel et al., 2010), (2) refléter l'activité de l'apprenant en lui offrant la possibilité de visualiser son activité (*mirroring*) (Jermann et al., 2001), (3) utiliser les traces pour guider l'apprenant dans son activité d'apprentissage en lui proposant par exemple l'action suivante à réaliser (Muñoz-Merino et al., 2010) (Kirschenmann et al., 2010), (4) assister les analystes-chercheurs en leur proposant des traces qui présentent généralement un niveau d'abstraction élevé permettant une analyse de la situation d'apprentissage en fonction des objectifs et hypothèses de recherche (Bratitsis et Dimitracopoulou 2006).

Dans (Settouti, 2011) un Système à Base de Traces Modélisées (SBT) est présenté comme une sorte de Système à Base de Connaissances, dont la source de connaissance est l'ensemble des traces d'interaction d'un utilisateur avec un système. Un SBT manipule des traces modélisées. Une trace modélisée est une trace munie d'un modèle exprimant la sémantique de son contenu.

Un SBT est défini dans (Settouti et al., 2007) comme : *« tout système informatique dont le fonctionnement implique à des degrés divers la gestion, la transformation et la visualisation de traces modélisées explicitement en tant que telles »*. Dans (Settouti, 2011), les concepts de base d'un SBT ont été définis formellement. Il s'agit des notions de modèle de trace, trace modélisée, schéma (pattern), requête et transformation. Un modèle de trace définit un vocabulaire pour décrire une trace. Il permet de (1) préciser la représentation du temps dans la trace, (2) classifier les éléments observés ainsi que les attributs qui les décrivent, et les éventuelles relations hiérarchiques entre ces classes d'observés (classe au sens du paradigme orienté objet), (3) définir les types de relations pouvant exister entre les observés, et les éventuelles relations hiérarchiques entre ces relations.

Une trace modélisée (M-Trace) est une trace, munie d'un modèle de trace. Elle définit une représentation du temps sur le domaine temporel défini par le modèle. Elle est composée d'un ensemble d'observés et de relations entre observés respectant les classes d'observés et de relations définis dans son modèle.

Un *pattern* permet d'exprimer un ensemble de critères à satisfaire par les éléments observés d'une trace modélisée. Il est possible d'exécuter des requêtes sur les traces en

utilisant la notion de pattern. Une requête est donc définie par un nom, un pattern qui défini les filtres à exécuter et l'ensemble de variables à retourner.

La transformation d'une M-Trace consiste à prendre en entrée une ou plusieurs M-Traces, chacune conforme à un modèle de trace, et à produire une nouvelle M-Trace conforme à un nouveau modèle de trace en exécutant un ensemble de règles de transformation. Une règle de transformation étant un couple (*pattern*, *template*). Le pattern servant à identifier l'ensemble des observés des traces sources auxquels le template (pattern défini sur la trace cible) sera appliqué pour produire les fragments de la trace cible.

Les usages des traces étant variés, leur utilisation en temps réel est parfois nécessaire. Pour ceci, deux classes de traces modélisées ont été identifiées, les traces hors ligne et les traces en ligne. Les résultats d'une requête s'exécutant sur une trace en ligne doivent être réévalués à chaque fois que de nouveaux événements surviennent. Cette requête est dite continue.

La Figure 2 présente l'architecture d'un système à base de traces modélisées. Un système de traçage commence par collecter les traces des interactions. Le système de traçage construit alors des traces modélisées dites primaires souvent d'un niveau d'abstraction bas. Ces traces peuvent être capturées en temps réel à l'aide de sources de traçage actives et stockées dans un entrepôt actif, ce sont des M-Traces en ligne. Elles peuvent sinon être des M-Traces collectées en différé et être stockées dans un entrepôt persistant. Le système de transformation exécute des transformations sur les traces en appliquant des filtres, réécrivant ou agrégeant des éléments de traces. Ceci peut donner lieu à des M-Traces plus pertinentes pour une utilisation plus spécifique. Le système d'interrogation permet d'interroger la base de traces modélisées pour extraire des informations spécifiques nécessaires à une étude donnée.

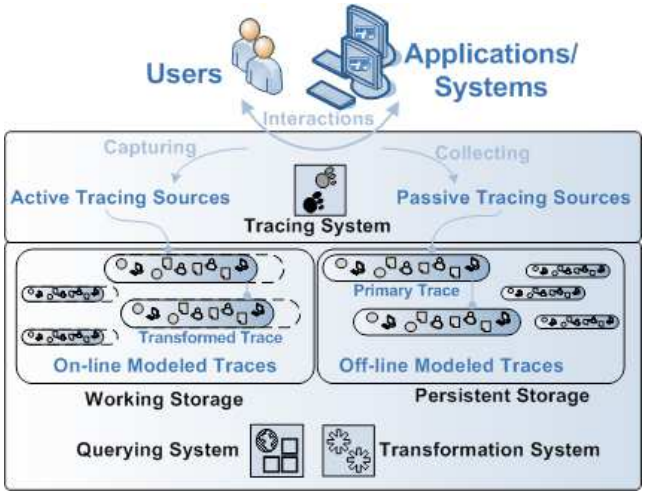


Figure 2 Architecture d'un Système à Base de Trace (Settouti et al., 2009)

Le SBT tel qu'il est formalisé est générique et peut être utilisé pour la gestion et la manipulation de traces d'interaction dans n'importe quel domaine d'application, et non pas seulement dans le domaine des EIAH. Ce cadre conceptuel a permis l'implémentation de systèmes dans des contextes spécifiques en proposant des modèles de traces et des transformations adéquats. SBT-IM (Djouad et al., 2010) et TREAM (Settouti et al., 2010), présentés dans les deux paragraphes suivants, sont deux exemples d'application de ce cadre conceptuel dans le domaine de l'apprentissage collaboratif et de l'apprentissage individuel respectivement. ABSTRACT (Georgeon et al., 2007) est une autre application de SBT et propose un outil d'ingénierie des connaissances à partir des traces d'activité, destiné aux ergonomistes, et dédié à l'analyse et la modélisation de l'activité dans le domaine de la conduite automobile. Il semble cependant difficile d'implémenter un système générique qui soit utilisable dans n'importe quel domaine et contexte, du fait des différences au niveau des traces manipulées dans les différents domaines. Le cadre conceptuel des SBT définit formellement un méta-modèle générique permettant la modélisation d'une trace mais ne propose aucun modèle ou format concret pour décrire le contenu d'une trace. Ce travail reste dans un niveau théorique et doit être implémenté selon les besoins spécifiques de chaque système.

2.2.3 SBT-IM

SBT-IM (Djouad et al., 2010) est un Système à Base de Traces dédié au calcul d'indicateurs pour les situations d'apprentissage collaboratives. Il est appliqué au calcul des indicateurs d'interaction dans la plateforme collaborative d'apprentissage Moodle (Moodle, 2013). Ce système est décomposé en trois sous-systèmes. Le premier construit une trace

première à partir des données collectées de la plateforme d'apprentissage, le deuxième permet la transformation des M-Traces et le troisième calcule des indicateurs à partir des M-Traces transformées. Dans (Djouad, 2008), un modèle d'utilisation d'une trace première générée par Moodle a été proposé (cf. Figure 3). Il est exprimé sous forme d'une ontologie OWL définissant les différents concepts pouvant exister dans une trace première. Ce modèle a été utilisé pour construire les M-Traces premières à partir des traces brutes de Moodle.

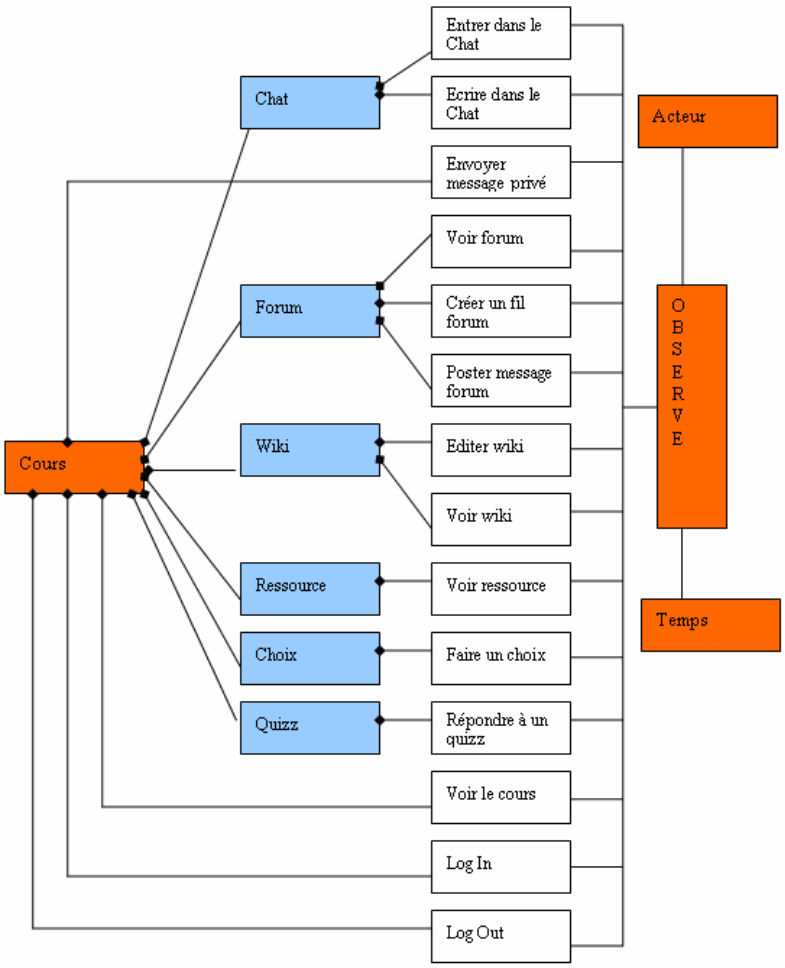


Figure 3 Ontologie pour le modèle d'utilisation de la trace

Cette ontologie est basée sur cinq concepts : les observés, l'acteur, le cours, le temps et les outils. Elle exprime les actions possibles par un acteur sur un outil à un instant donné. Un exemple de trace première qui instancie ce modèle peut être : « l'observé 'écrire message Chat1' est relevé dans la ressource 'Chat Public1' qui appartient au 'cours1' par l'acteur 'Bernard' à l'instant 't1' ».

Dans (Djouad, 2008), une première tentative d'export de la trace première exprimée suivant l'ontologie proposée vers le modèle des traces du corpus MULCE (qui sera présenté

dans la suite) (Reffay et al., 2008), a été réalisée. En effet, l'un des objectifs de la modélisation de la trace et de sa gestion indépendante de l'environnement de l'apprentissage est sa réutilisation selon les différents besoins des chercheurs. Des opérateurs de transformation définis par le SBT permettent ainsi d'exporter les traces premières modélisées vers d'autres modèles de traces selon les besoins.

Ce travail est une application concrète de l'utilisation de l'architecture SBT pour l'analyse de situations d'apprentissage collaboratives. Il permet de construire des traces modélisées, de les transformer et de les utiliser pour calculer des indicateurs sur l'activité permettant la personnalisation du fonctionnement de l'environnement d'apprentissage. Ce système permet aussi, conformément au cadre théorique des SBT (Settouti et al., 2007), de mettre les traces sous une forme particulière pour les rendre utilisables dans d'autres systèmes (comme l'exemple de MULCE).

2.2.4 Trace-Based Learner Modeling (TREAM) Framework[5]

Parmi les aspects importants de la personnalisation des EIAH, on trouve la modélisation de l'apprenant permettant l'adaptation du fonctionnement de l'EIAH au profil de l'apprenant en question. Le travail présenté dans (Settouti et al., 2010) s'intéresse à la modélisation d'un apprenant à partir des traces de son interaction avec un EIAH durant une session d'apprentissage. Ce travail se veut générique et le framework proposé est donc conçu pour être utilisable par différents EIAH offrant différents types d'apprentissage individuel. Ce travail étend le cadre théorique des Systèmes à Base de Traces (Settouti, 2011), en définissant de nouveaux modèles de connaissance permettant la construction de profils d'apprenants.

La Figure 4 ci-dessous illustre le fonctionnement de TREAM qui s'effectue en trois étapes pour construire le profil d'apprenant à partir des traces d'une activité d'apprentissage individuelle : (1) La première étape consiste en la construction de la trace première modélisée correspondant aux interactions de l'apprenant avec l'EIAH. Les notions de trace modélisée et de modèle de trace sont reprises du travail accompli sur les SBT dans (Settouti et al., 2009). Cependant, comme on l'a déjà signalé, ce travail n'ayant pas offert de modèle pour

[5] Ce travail est effectué dans le cadre de la tâche 3 « Interprétation des traces et représentation des connaissances » du projet ISLE/EIAH de la région Rhône-Alpes.

représenter le contenu de la trace d'apprentissage, TREAM propose le modèle LATM (Learning Activity Trace Model) pour définir le vocabulaire des éléments pouvant constituer une trace d'apprentissage individuelle issue de l'interaction d'un apprenant avec un EIAH. Un modèle de trace est défini dans le SBT par l'ensemble des classes d'observés, de leurs attributs, et des éventuelles relations sémantiques entre observés. Le modèle LATM définit d'une manière générique et extensible l'ensemble des classes d'observés et les relations qui les lient. Une représentation RDF (RDF, 2004) du modèle LATM est représentée dans la Figure 5 ci-dessous. Pour construire la trace première modélisée, spécifique à un EIAH donnée, le concepteur du système décrit le modèle $LATM_{TEL}$ (cf. Figure 4) qui définit le vocabulaire de la trace issue de cet EIAH. Ce modèle peut être une réutilisation ou une extension du modèle LATM générique proposé dans le *framework* (Figure 5). Ce modèle est par la suite utilisé pour associer à chaque observé collecté par l'EIAH le concept ou la relation correspondante donnant lieu ainsi à la trace première modélisée ; (2) La deuxième étape consiste à faire les transformations nécessaires sur la trace première si celle-ci est dans un niveau d'abstraction bas et nécessite des abstractions pour la rendre plus significative, ou si elle contient un grand nombre d'informations dont certaines ne sont pas utiles pour l'analyse ; (3) La troisième et dernière étape utilise les traces modélisées premières ou transformées pour inférer le profil de l'apprenant. À côté des traces modélisées, le framework fournit des informations complémentaires exprimées sous forme d'un modèle de l'apprenant et d'un modèle de diagnostic. Ces deux modèles ont été définis d'une manière générique. Le modèle de l'apprenant permet au concepteur de définir les concepts abstraits utiles pour construire le profil de l'apprenant, comme les informations relatives à l'apprenant, sa stratégie de résolution et ses connaissances. Le modèle de diagnostic permet au concepteur d'exprimer les connaissances nécessaires pour la résolution d'un problème comme le plan de travail, les solutions, les procédures et éventuellement les bugs. Il permet aussi d'exprimer des règles et des requêtes nécessaires pour effectuer le diagnostic. Dans (Settouti et al., 2011), des patrons de requêtes réutilisables exprimées en SPARQL (SPARQL, 2008) sont proposés afin de permettre la détection d'éléments de profils d'apprenants à partir des traces.

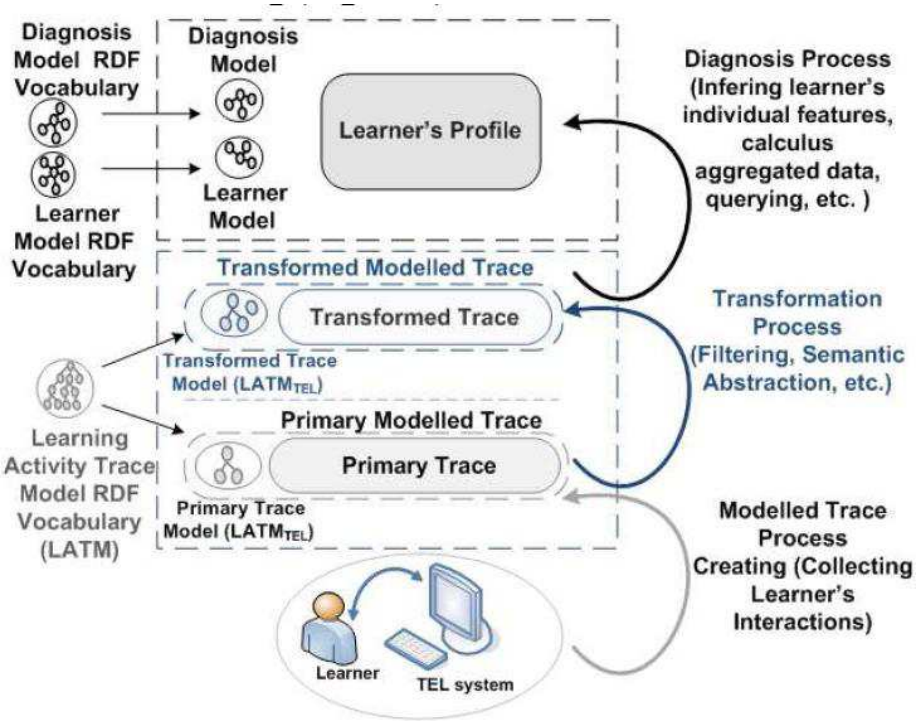


Figure 4 Trace-Based Learner Modeling (TREAM) Framework (Settouti et al., 2010)

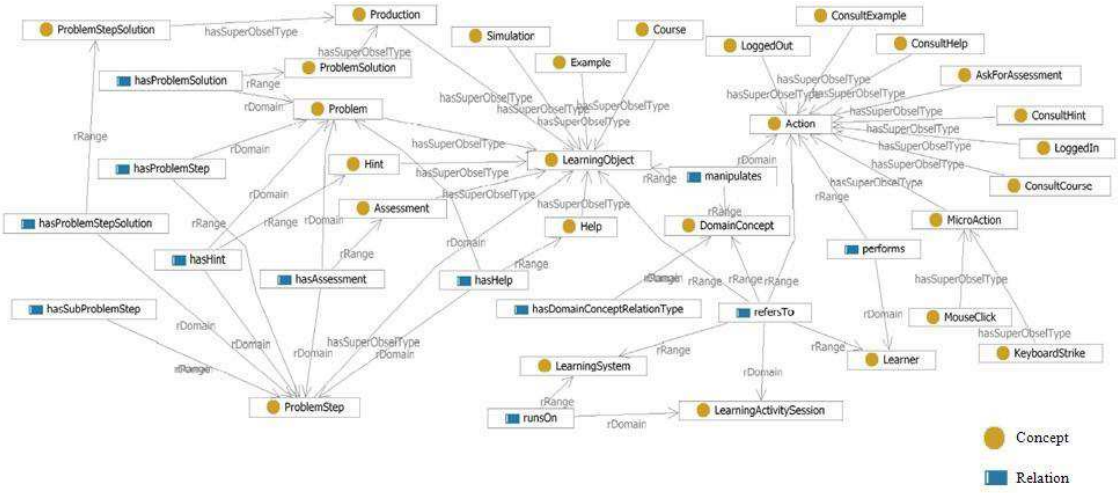


Figure 5 Modèle RDF de la trace d'activité d'apprentissage individuel (LATM) (Settouti et al., 2010)

Ce travail illustre encore une fois l'utilité potentielle d'un système à base de traces dans la personnalisation d'un EIAH. Un *framework* pour la construction de profil d'apprenant à partir de traces d'interaction modélisées à été proposé. Les traces doivent cependant être complétées par des informations sur le domaine d'application. Ce travail a encore besoin d'être implémenté et testé sur des traces provenant de différents EIAH.

2.2.5 TATIANA

TATIANA (*Trace Analysis Tool for Interaction ANAlysts*) ((Dyke, 2009), (Dyke et al., 2008)) est un outil destiné aux chercheurs intéressés par l'analyse de situations d'interactions médiatisées par ordinateur. TATIANA permet d'analyser à posteriori les traces d'interaction issues d'une situation d'apprentissage, avec des corpus composées de ressources de différents types pouvant être des fichiers de log, audio/vidéo ou encore une transcription de vidéo.

Pour permettre l'analyse de ces traces, celles-ci doivent être importées dans un format lisible par TATIANA, appelé le « Display Format » ou « Tatiana Info File (TIF) ». Ce format représente une modélisation très simple et générique de la trace pouvant s'adapter à différents types de traces, qui est assez proche d'une modélisation clé-valeur de la trace. Il est essentiellement utilisé par Tatiana pour représenter les traces d'interaction, générées par des EIAH tels que DREW (Corbel et al., 2003) ou COFFEE (De Chiara et al., 2007), sous forme d'une séquence d'événements appelée « rejouable » (replayable). Une collaboration entre ce projet et le projet MULCE (Reffay et al., 2008) (cf. paragraphes 2.2.13 et 2.3.5) a également permis d'analyser les données partagées dans un corpus du projet MULCE.

Un « rejouable » est constitué d'un ensemble d'*items* associés aux différents événements formant la trace. Un item est construit par un ensemble de facettes dont la facette « time » obligatoire et permettant de situer l'événement dans le temps. Par ailleurs, un *item* peut aussi contenir un nombre variable de facettes nécessaires à la description d'un événement donné, comme par exemple l'outil ayant créé la trace, l'utilisateur concerné, le contenu de l'interaction, etc. Une DTD nommée « display.dtd » a été définie pour décrire les fichiers manipulés dans Tatiana dont les traces. Ce format est simple et permet de représenter d'une manière flexible les traces générées par un EIAH.

TATIANA, par sa séparation entre les traces d'interaction et les analyses construites à partir de ces traces, permet la capitalisation et le partage des analyses. Le processus d'analyse mis en place dans Tatiana est simple et itératif. En effet, comme présenté dans la Figure 6, le processus d'analyse commence par une question de recherche à étudier. Pour ce faire, il est nécessaire de collecter des données d'interactions qui constitueront le corpus initial de données dans la collection des données d'étude (*Data Pool*). Les données jusqu'alors collectées sont interprétées. Si l'interprétation fournit au chercheur les résultats attendus, le processus d'analyse est alors achevé et les résultats peuvent être publiés, sinon un nouvel

artefact d'analyse est créé et ajouté à la collection des données pour être interprété. Le processus d'analyse s'exécute en boucle jusqu'à obtention des résultats satisfaisants, il peut également être repris dans un but de rectification et de capitalisation des analyses.

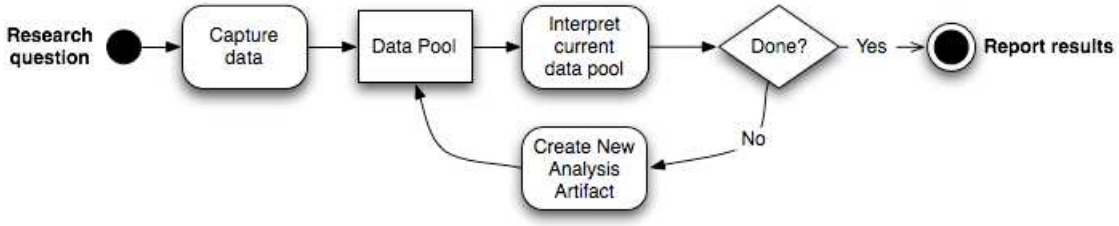


Figure 6 Modèle du processus d'analyse de TATIANA (Dyke, 2009)

TATIANA permet aux chercheurs d'analyser les corpus de traces d'interaction pour évaluer le déroulement d'une séance de travail médiatisée par ordinateur et tracée. La trace est représentée sous forme de « rejouable » qui peut être (1) rejoué, (2) synchronisé avec d'autres rejouables (par exemple synchroniser une vidéo avec sa transcription), (3) visualisé (sous forme d'un tableau ou d'une ligne graphique temporelle) , (4) transformé et (5) enrichi. Le modèle de la trace est simple et général ce qui permet de représenter différents types de traces mais rend difficile le développement de traitements automatiques spécifiques des traces qui nécessitent une modélisation plus détaillée de leur contenu.

2.2.6 CARTE

CARTE (*Collection, activities Analysis and Regulation based on Traces Enriched*) (Courtin, 2008) est une station d'observation de situations d'apprentissage médiatisées par ordinateur. Ce travail, bien que plus ancien, est en parfait accord avec le modèle théorique des Systèmes à Base de Traces modélisées. La station d'observation a pour rôle de collecter des traces d'interaction et d'effectuer des transformations sur ces traces et ceci d'une façon indépendante des outils utilisés dans la réalisation d'une séance d'apprentissage collaboratif (par exemple un outil de chat pour les discussions entre apprenants, un éditeur de texte partagé pour les productions collaboratives).

La collecte des traces se fait par l'instrumentation des outils utilisés dans une session d'apprentissage, à l'aide d'une API (*Application Programming Interface*) de collecte n'obligeant pas les outils à modifier leur fonctionnement de base. Les traces collectées sont utilisées pour analyser l'activité d'apprentissage et réguler l'activité en temps réel, mais peuvent aussi faire l'objet d'analyses en temps différé pour évaluer une session

d'apprentissage par un chercheur souhaitant répondre à une question de recherche particulière. L'analyse des traces collectées peut être faite dans CARTE à l'aide d'un outil appelé « Analyseur » permettant au chercheur d'exprimer des règles pour interpréter les traces brutes ou des traces déjà transformées afin d'expliciter des connaissances d'un niveau d'abstraction plus élevé que celui des traces brutes. L'analyse des traces peut aussi être réalisée à l'aide d'autres outils d'analyse et/ou de visualisation indépendants. Cette utilisation de différents outils distincts de la station d'observation suppose une interopérabilité entre les formats de données gérées par les différents outils.

Le format de trace géré par CARTE est schématisé dans la Figure 7 ci-dessous. Il permet d'exprimer des signaux et des séquences. Un signal représente un événement ponctuel associé à une date donnée. Une séquence peut être constituée de signaux et/ou séquences et possède une date de début et une date de fin. Un signal peut être construit à l'aide de l'API de collecte et représente donc une trace brute, ou peut être généré par une règle d'analyse exprimée par l'outil analyseur. Une séquence est toujours générée par l'analyseur. Afin d'être le plus générique possible, le format CARTE spécifie un ensemble de métadonnées considérées génériques pour la description d'un élément de trace. Un signal est décrit par : (1) la source qui l'a produit, cette source est soit l'outil ayant généré la trace brute, soit l'analyseur si c'est un signal ou une séquence construit par une règle d'analyse ; (2) sa description textuelle ; (3) sa date, qui indique quand l'événement est survenu ; (4) l'identifiant de l'événement (les événements possibles varient d'un outil à un autre) ; et (5) l'outil dans lequel a eu lieu l'événement. Une séquence est décrite par : (1) la source qui est généralement l'analyseur ; (2) sa description textuelle ; (3) sa date de début ; (4) sa date de fin ; et (5) un identifiant du type de la séquence.

Conformément à la définition formelle de la trace modélisée (Settouti, 2011), une trace CARTE est associée à un modèle d'utilisation expliquant la sémantique des éléments collectés dans la trace, ceci étant exprimé par le biais de paramètres. Les traces collectées dans CARTE sont ensuite utilisées pour fournir aux participants des informations qui les intéressent sur l'activité des autres participants en temps réel (par exemple informer un tuteur des activités des apprenants). Une autre utilisation des traces consiste en la synchronisation des outils logiciels entre eux (par exemple, un apprenant ne peut pas joindre un groupe auquel il n'a pas été inscrit). Ces retours du système d'observation sont réalisés suite à l'évaluation, par l'analyseur, d'un ensemble de règles prédéfinies. Les traces peuvent aussi être utilisées en différé, pour une analyse globale de l'activité.

La modélisation de la trace dans ce projet est assez générique, elle permet donc de représenter des traces provenant de différents EIAH mais présente des difficultés d'utilisabilité quant à l'automatisation de l'exploitation des traces. En effet, vu que les spécificités des systèmes sont collectées dans les paramètres, l'exploitation de ces paramètres est liée au système considéré.

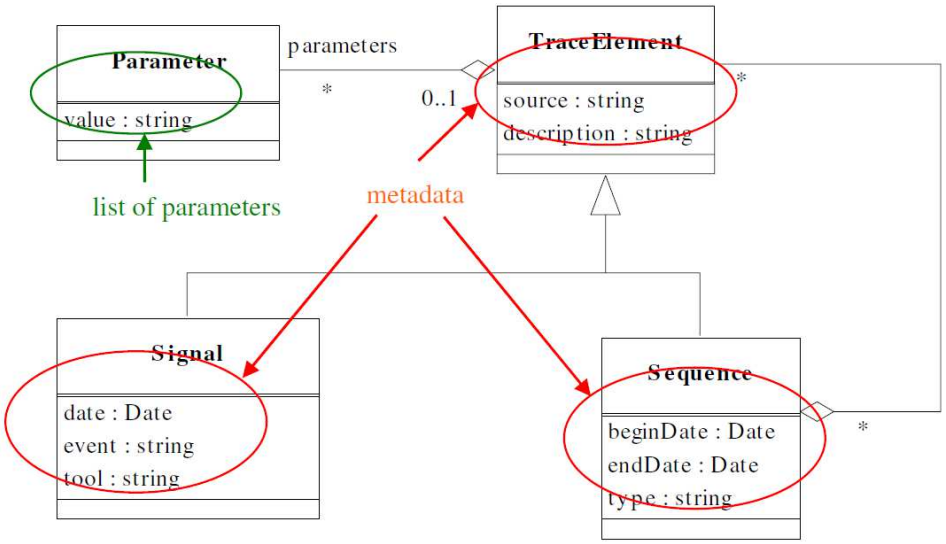


Figure 7 Modèle UML d'une trace dans CARTE (Courtin et Talbot, 2009)

2.2.7 TrAVis et les communications médiatisées par ordinateur[6]

TrAVis (Tracking Data Analysis and Visualization) (May, 2009) est un outil de collecte, d'analyse et de visualisation en temps réel permettant le calcul d'indicateurs à partir des traces d'utilisation des outils de communication tels que les forums et chat. Cet outil permet d'assister les apprenants et les tuteurs impliqués dans des situations d'apprentissage à distance. En effet, cet outil permet au tuteur de suivre en temps réel les activités des apprenants, il permet également de donner des feedbacks aux apprenants sur leur activité durant une session d'apprentissage.

Un travail préalable à la conception et la mise en œuvre de l'outil TrAVis a été mené pour étudier la possibilité de tracer exhaustivement les interactions provenant de l'utilisation des outils de communication par un utilisateur engagé dans une situation d'apprentissage à distance (May et al., 2008). Afin de collecter des traces qui soient les plus complètes possible,

[6] Ce travail est fait dans le cadre de la tâche 4 « Interprétation des traces et régulation des interactions sociales et langagières » du projet ISLE/EIAH de la région Rhône-Alpes.

l'approche proposée (cf. Figure 8) repose sur deux types de collecteurs de traces, l'un du côté du client et l'autre du côté du serveur. Quatre types d'interaction ont été identifiés : (1) HHIMC, interaction Homme-Homme médiatisée par ordinateur ; (2) MMI, interaction Machine-Machine ; (3) HCI, interaction Homme-Machine ; et (4) CA, action de l'ordinateur. Les collecteurs implémentés prennent en considération uniquement les HHIMC et HCI. Un ensemble de modèles d'utilisation est utilisé par les collecteurs de traces. Ces modèles décrivent de quelle manière toute activité de communication peut être effectuée par un utilisateur, et comment représenter les traces des actions/interactions des utilisateurs et les contenus associés des communications.

Les traces d'utilisation des outils de communication peuvent différer selon les objectifs qui motivent leur collecte, ce qui entraîne un problème de réutilisabilité dû aux différents formats utilisés. Afin de pallier ce problème, l'idée d'exprimer un modèle générique pour représenter les traces de communication médiatisée par ordinateur (CMC pour Computer Mediated Communication) s'impose (May et al., 2008). En effet, ce modèle permet de : (1) proposer une représentation commune des traces CMC ; (2) transformer une trace dans différentes représentations (par exemple Base de données relationnelle et fichier XML) tout en respectant toujours le même modèle de traces ; et (3) enrichir des traces CMC existantes par des informations supplémentaires à celles de la représentation originale. Cette proposition a pour objectif d'augmenter l'utilisabilité des traces de communications.

La Figure 9 illustre la représentation graphique du schéma XML du modèle générique proposé dans (May, 2009) pour la représentation des traces CMC. Ce modèle propose une structure générique d'une trace CMC. En effet, cette structure est conçue pour représenter toute trace CMC indépendamment de l'outil de communication utilisé (forum, chat, wiki, etc.). Une trace CMC est modélisée par un ensemble d'éléments tels que l'utilisateur, l'estampillage temporel, le contenu de l'interaction, etc. Pour être flexible, le modèle contient un élément « attribut » permettant de représenter des informations supplémentaires pour décrire et spécifier une activité de communication.

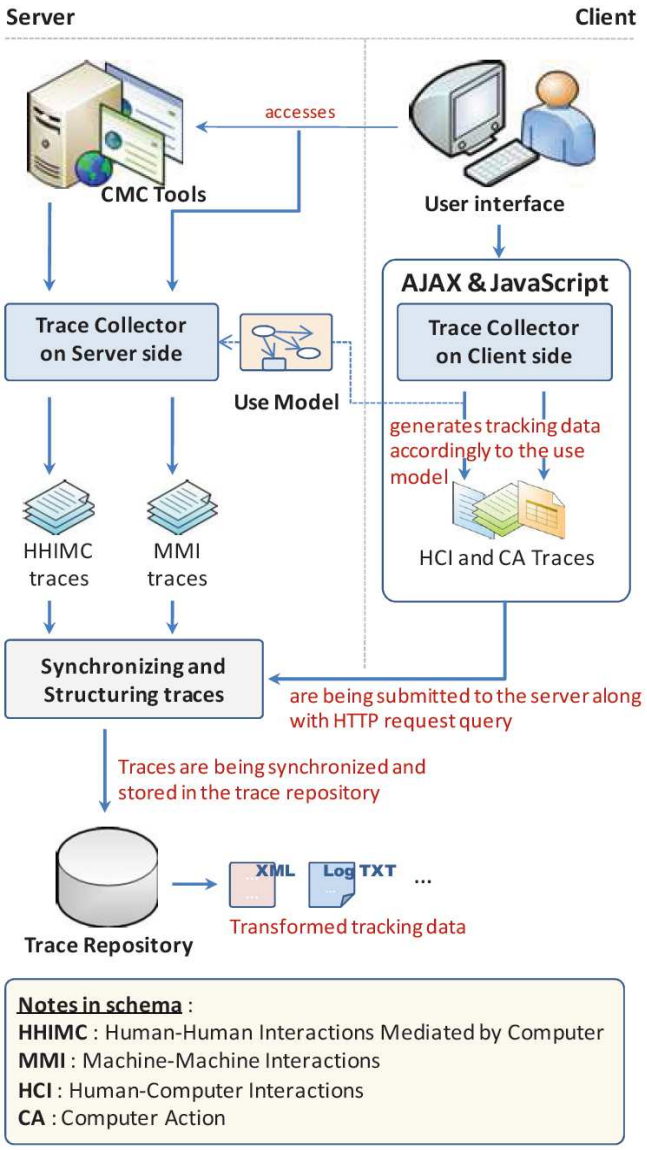


Figure 8 Architecture web du système de traçage côtés client et serveur (May et al., 2008)

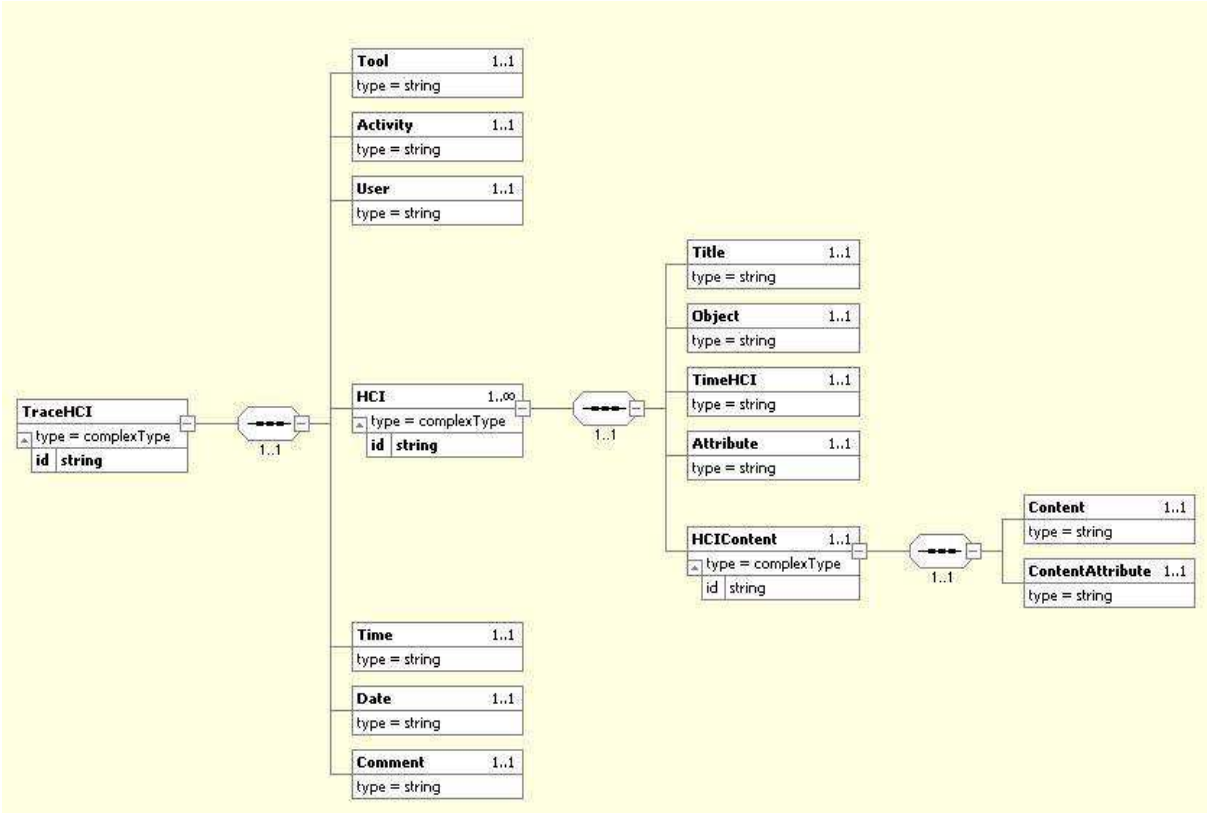


Figure 9 Modèle des traces CMC (Computer Mediated Communications) (May, 2009)

2.2.8 CAM

CAM (Contextualized Attention Metadata) (Wolpers et al., 2007) est un framework qui s'intéresse à la collecte des métadonnées d'attention contextualisée pour pallier le problème de surcharge d'information auquel sont confrontés les enseignants et apprenants dans leur travail de tous les jours. Les CAM sont des données qui concernent la concentration de l'attention et l'activité d'un utilisateur. Elles permettent de décrire des informations telles que : quels objets attirent l'attention des utilisateurs ; quelles actions exécutent les utilisateurs sur ces objets, et quels contextes d'utilisation. On peut conclure que les CAM constituent une trace spécialisée : des données collectées pour enregistrer ce qui se passe quand un utilisateur interagit avec les objets d'un système. La collecte des CAM permet la personnalisation des environnements d'apprentissage en proposant aux utilisateurs des objets et tâches optimisés qui s'accordent le mieux avec leur profil, mais aussi en assistant la composition des groupes d'utilisateurs dans un contexte collaboratif.

Un schéma de représentation des CAM est proposé. Ce schéma a été conçu pour être général et permettre la représentation des traces d'activité des utilisateurs dans tous types de systèmes qu'ils utilisent durant leur travail. Il est exprimé sous forme d'un schéma XML et

étend le schéma proposé dans Attention.XML (Çelik, 2005). Ce dernier est conçu pour collecter des observations des utilisateurs liées à des activités de navigation et de consultation d'informations dans des blogs et des fils de nouvelles RSS (RSS, 2009). La Figure 10 ci-dessous illustre le schéma CAM. Toutes les interactions d'un utilisateur avec tous les systèmes sont collectées dans l'élément « group » qui comprend des éléments « feed ». Un élément « feed » regroupe toutes les interactions d'un utilisateur avec un système particulier. L'élément « item » collecte l'attention donnée à un document numérique spécifique, et est lié aux événements dans lesquels le document est impliqué par des éléments de type « event ». Le schéma proposé a été utilisé pour collecter des traces produites dans le cadre d'une situation de travail provenant du logiciel « Microsoft Powerpoint », du navigateur Web « Mozilla Firefox », de l'outil de messagerie instantanée « MSN Messenger », et du lecteur multimédia « Winamp ». Les différentes traces ont été structurées d'une manière homogène et stockées dans une base de données XML centralisée. Dans le cadre du projet ROLE[7] (Responsive Open Learning Environments), le schéma CAM a été utilisé et une API a été proposée pour la collecte des CAM et leur stockage centralisé. Par ailleurs, le projet offre également des services Web permettant le stockage et l'interrogation des données sur un serveur de partage.

Ce travail a comme objectif de collecter des données sur l'attention provenant de différents systèmes afin d'aider les enseignants et apprenants à mieux digérer le flux des informations disponibles. Les données collectées peuvent être utilisées pour dégager des statistiques sur l'utilisation des documents d'apprentissage, identifier et extraire des *patterns* de comportement permettant de classifier les utilisateurs et de les conseiller en fonction de l'expérience passée. Malgré l'intérêt que présente cette initiative par la proposition d'une représentation unique des traces d'interaction provenant de différents systèmes informatiques, la plupart des projets redéfinissent un modèle pour les traces qu'ils collectent pour s'adapter à la spécificité de leur domaine d'application. En effet, comme conclut (Bolchini et al., 2007), l'utilisabilité et la généralité du modèle sont inversement proportionnelles, c'est-à-dire, plus le modèle est expressif et général, moins il est pratique et utilisable.

Dans (Butoianu et al., 2010), une modélisation des CAM basée sur le standard CIM (Common Information Model) (CIM, 2013) est proposée. CIM est un standard du consortium DMTF (Distributed Management Task Force) qui fait partie de l'initiative WBEM (Web-

[7] http://www.role-project.eu/

Based Enterprise Management) qui propose un ensemble de standards et de technologies offrant des solutions de gestion des environnements réseaux distribués. CIM définit trois types de modèles : modèle de base qui définit des données de gestion génériques, modèle commun qui étend un modèle de base et définit des données de gestion partagées pour un domaine de gestion spécifique, et modèle d'extension permettant d'étendre un modèle commun avec des données liées à l'environnement technologique. Dans le travail précité, les chercheurs ont étendu les modèles de base définis dans le standard CIM pour représenter les CAM en fonction des besoins d'observation et des environnements d'apprentissage. Ils ont défini deux modèles génériques. Le premier permettant de représenter les données génériques relatives aux systèmes d'apprentissage et des ressources, alors que le deuxième représente les interactions des utilisateurs avec ces systèmes et ressources. Des modèles spécifiques ont ensuite été définis pour représenter les données relatives à des environnements d'apprentissage et ressources spécifiques, et aux activités des utilisateurs. La Figure 11 ci-après illustre l'architecture de la plateforme proposée pour collecter, stocker, et interroger les CAM tout en protégeant les données privées. C'est une architecture composée de trois couches. La première couche « contexte d'apprentissage » permet la collecte des données au niveau de l'environnement d'apprentissage. La deuxième couche « *middleware* » offre un ensemble de services web facilitant la communication entre les deux autres couches, en l'occurrence la conversion des données collectées par les agents de la première couche pour être traitables par la troisième couche, la manipulation des modèles ainsi que l'interrogation des données stockées dans la troisième couche. Enfin, la troisième couche « contexte d'observation » permet le stockage des CAM collectées suivant les modèles construits. Cette proposition est intéressante car elle tente de capitaliser sur les données génériques tout en gardant la possibilité de spécifier les modèles permettant ainsi une meilleure utilisabilité. Les données collectées par l'agent de l'environnement d'apprentissage pouvant toutefois avoir un modèle différent de celui défini dans la couche « contexte d'observation », certaines données contenues dans les traces d'origine peuvent être perdues au moment de la conversion. Par ailleurs, l'un des objectifs énoncés de ce travail étant le partage des CAM et étant donné que la description du contexte de collecte de ces CAM n'est pas décrit, nous pensons que cet objectif ne peut être atteint que pour des chercheurs d'une même communauté ayant une idée sur le contexte d'apprentissage ayant donné lieu aux données collectées.

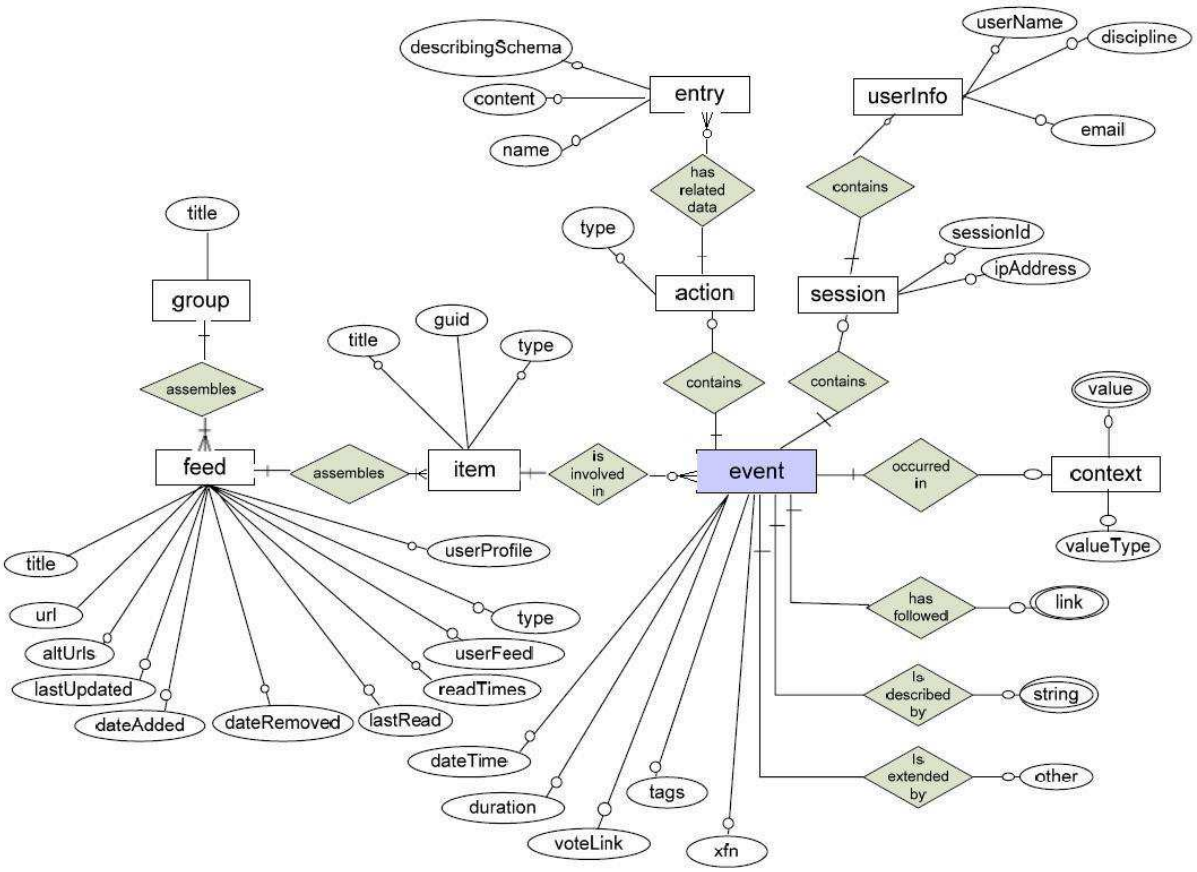


Figure 10 Eléments du schéma CAM (Wolpers et al., 2007)

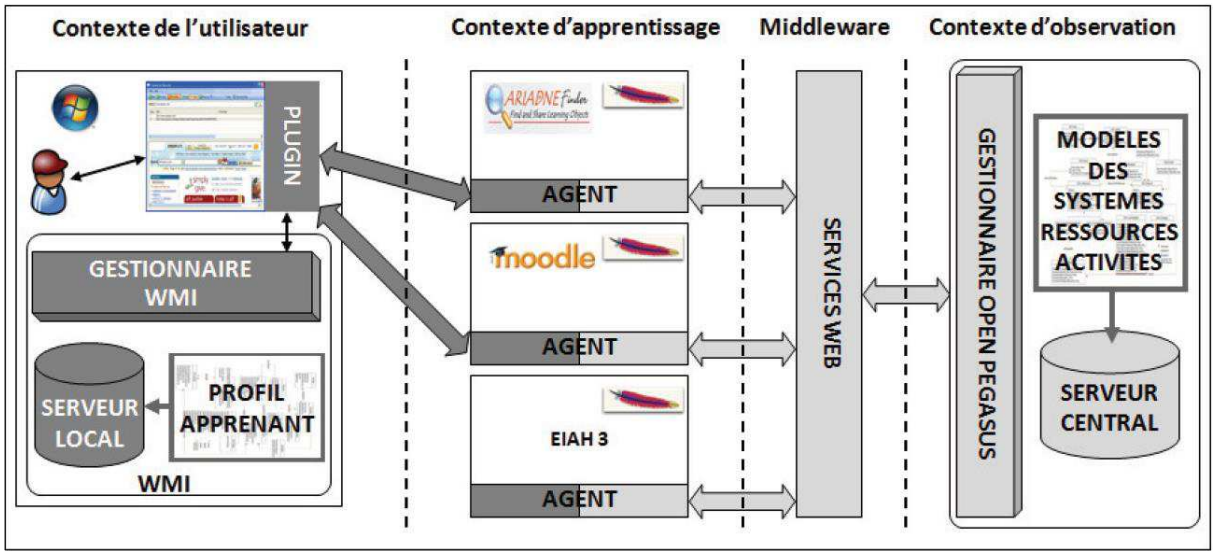


Figure 11 Architecture basée sur le standard CIM pour la collecte et le partage des CAM (Butoianu et al., 2011)

2.2.9 UICO

UICO (User Interaction Context Ontology) (Rath et al., 2009) est une ontologie proposée dans le cadre du projet DYONIPOS (Granitzer et al., 2009) qui s'intéresse à la

problématique de détection du contexte des utilisateurs. Dans ce projet, les systèmes à base de traces sont désignés par les systèmes « conscients » du contexte (*context-aware systems*). La trace d'interaction est quant à elle désignée par le « contexte d'interaction de l'utilisateur ». Le contexte est défini comme « toutes les interactions de l'utilisateur avec les ressources, les applications et le système d'exploitation de son bureau d'ordinateur ». Dans ce projet, le choix s'est porté sur la modélisation ontologique du contexte suivant les recommandations des études des différents types de modélisation de contexte publiées dans (Strang et Linnhoff-Popien, 2004) et (Baldauf et al., 2007). Ces deux dernières études ayant analysé les différents types de modélisation du contexte : modèles clé-valeur, modèles de schémas de balisage, modèles graphiques, modèles orientés-objets, modèles logiques, et modèles ontologiques ; elles ont conclu que la modélisation ontologique du contexte est la plus prometteuse grâce à la flexibilité, l'extensibilité, et l'expressivité qu'elle offre.

La Figure 12 ci-dessous illustre la représentation conceptuelle « pyramide sémantique » proposée par le projet pour représenter les relations entre les événements, les blocs d'événements et les tâches permettant la détection du contexte d'interaction de l'utilisateur. En bas de la pyramide se trouvent les événements provenant des interactions de l'utilisateur avec les applications de son bureau d'ordinateur. Plus haut se trouvent les blocs d'événements qui sont des séquences d'événements liés logiquement, un bloc d'événements reliant les actions de l'utilisateur sur une ressource spécifique. Au sommet de la pyramide se trouvent les tâches qui regroupent des blocs d'événements et qui représentent des étapes indivisibles bien définies d'un processus et impliquant un seul utilisateur.

La Figure 13 ci-dessous illustre l'ontologie UICO modélisée en OWL (OWL, 2004) et composée de 107 classes et 281 propriétés (dont 224 propriétés de types de données et 57 propriétés d'objets). Quatre dimensions composent l'ontologie UICO : (1) la dimension *action*, composée des concepts qui représentent les actions des utilisateurs et les états des tâches ; (2) la dimension *ressource*, composée des concepts qui représentent les ressources disponibles par les applications du bureau d'ordinateur ; (3) la dimension *application*, une dimension cachée, représentée par deux propriétés du concept « event » permettant de collecter des informations sur l'application avec laquelle l'utilisateur interagit ; et (4) la dimension « utilisateur » composée de concepts permettant la description de l'utilisateur et sa session.

Le projet UICO soutient la position que la modélisation du contexte doit être spécifique à un domaine d'application ce qui augmente son utilisabilité. Il propose une modélisation ontologique répondant à des besoins d'observation permettant la détection du contexte d'interaction d'un utilisateur avec les outils de son bureau d'ordinateur, avec comme objectif ultime la détection de tâches permettant de développer des outils personnalisés. Ce choix ne fait cependant pas l'unanimité puisque ceux qui prônent la généralité trouvent que ce genre de solution permet rarement la réutilisation.

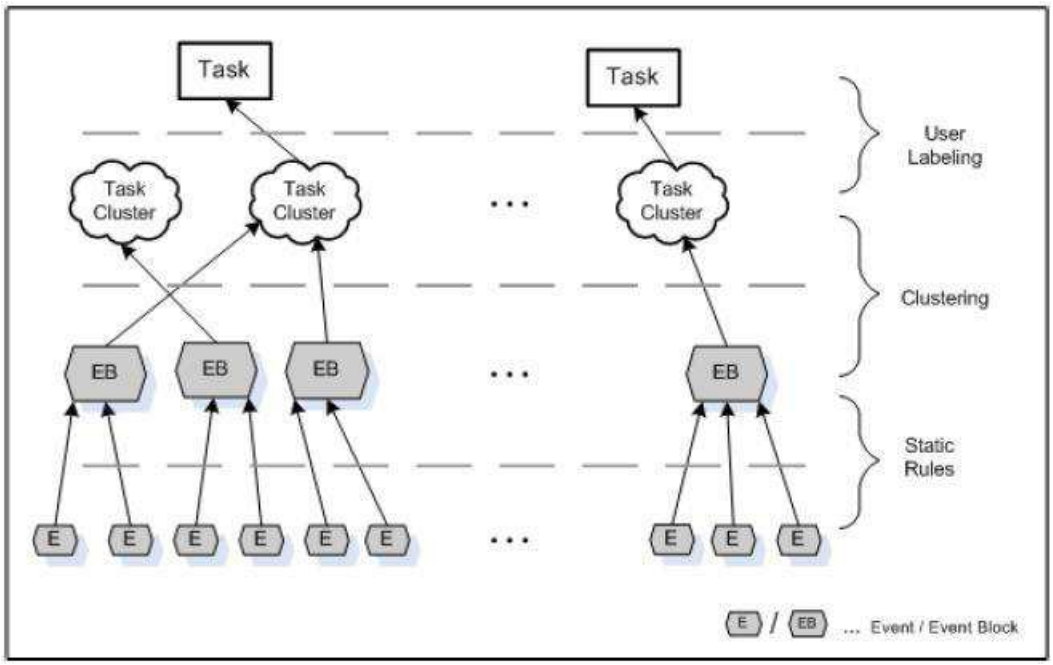


Figure 12 Pyramide sémantique: représentation conceptuelle des relations entre événements, blocs d'événements, et tâches pour la détection du contexte

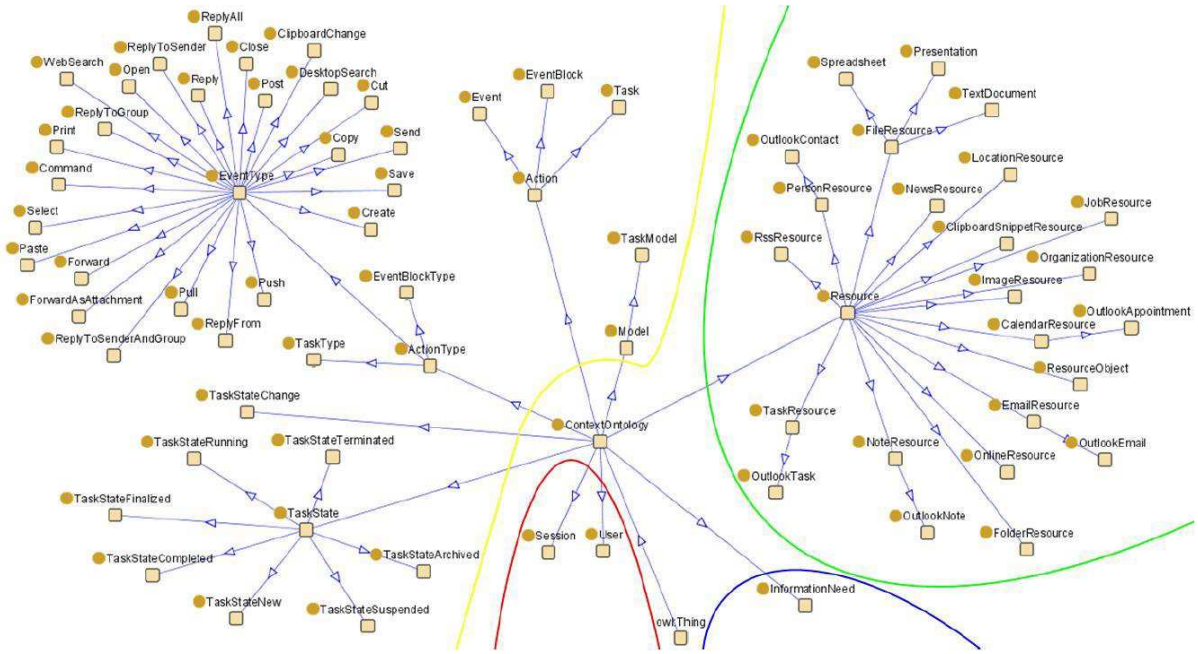


Figure 13 Les concepts des quatre dimensions de l'ontologie UICO

2.2.10 Common Format, IA JEIRP Kaleidoscope

IA (*Interaction Analysis Supporting Teachers & Students' Self-regulation*) est un projet JEIRP (*Jointly Executed Integrated Research Project*) du réseau d'excellence européen Kaleidoscope. La tâche 2 du projet intitulée « Interaction Analysis Methods » (Martínez et al., 2005) s'intéresse à l'étude des méthodes d'analyse des interactions. L'objectif de cette tâche était de construire une librairie d'outils d'analyse des interactions dans le domaine des EIAH. La Figure 14 ci-dessous illustre la situation initiale du projet : neuf environnements d'apprentissage, ayant chacun son propre format de trace, fortement couplés à sept outils d'analyse. L'objectif est de fournir les outils d'analyse dans une librairie permettant leur utilisation dans l'analyse des interactions issues de différents environnements d'apprentissage, et non seulement de l'EIAH auquel il est fortement couplé.

Afin d'atteindre cet objectif, la solution proposée est de définir un format commun (Martínez et al., 2005) pour représenter les traces d'interactions générées suite à l'utilisation des environnements d'apprentissage. Ce travail s'inspire d'un travail antérieur (Martínez et al., 2003) ayant proposé un format XML pour représenter les traces d'interactions dans les environnements d'apprentissage collaboratifs. Ce format doit permettre la structuration des données de manière flexible et standardisée, et doit être adaptable à différentes perspectives d'analyse et à différentes situations collaboratives.

La Figure 15 ci-dessous illustre une partie de la DTD du format commun proposé. Ce format doit fournir une structure contenant toutes les données utiles aux différents outils d'analyse partagés dans la librairie. En définissant ce format, un compromis doit être trouvé entre un format bien défini et structuré et un format flexible. Pour cela, il convient de trouver une partie commune obligatoire pour exprimer les informations minimales requises, et de laisser la possibilité d'ajouter des informations supplémentaires optionnelles au besoin, suivant les données générées par les environnements d'apprentissage, et pouvant être utiles pour certaines techniques d'analyse.

Une trace d'interaction est définie grâce à deux parties : (1) une première partie « Preamble » qui décrit le scénario pédagogique du déroulement de la situation d'apprentissage tracée, cette partie donne une idée sur la configuration générale de la situation d'apprentissage, par exemple des informations sur les participants et leurs rôles, les sous-groupes, les ressources externes disponibles, etc. Ces informations favorisent la compréhension du scénario pédagogique et des processus d'analyse adaptés ; (2) une partie « Actions » qui concerne l'ensemble des actions tracées et décrites par des propriétés dont certaines sont obligatoires et d'autres sont optionnelles. Parmi ces propriétés, on trouve le timestamp, le type de l'action, le ou les utilisateurs impliqués, le contenu qui décrit l'action (par exemple le message envoyé dans un chat) et les objets manipulés dans la réalisation de l'action.

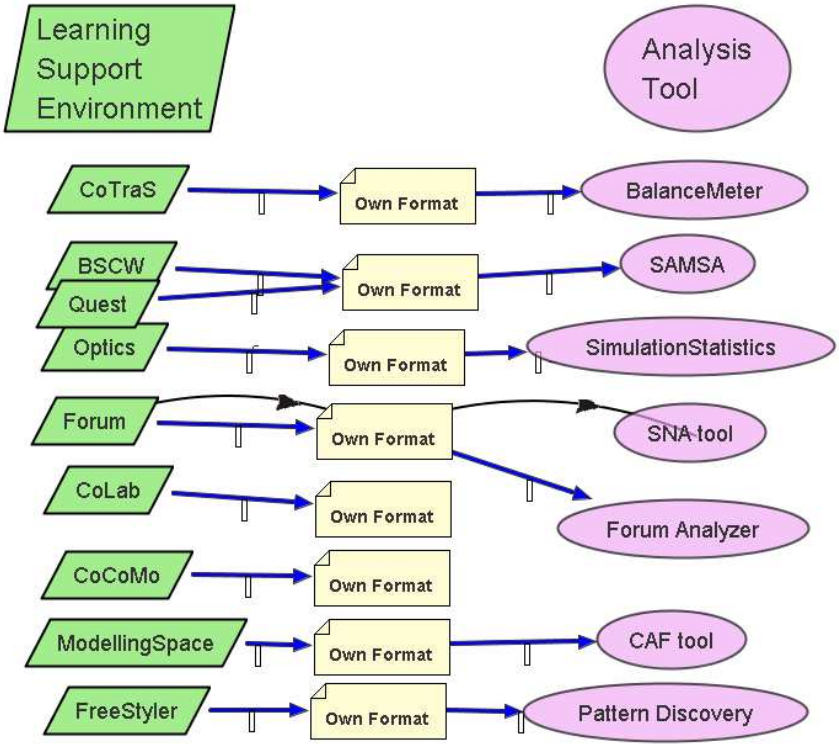


Figure 14 Situation initiale - outils d'analyse fortement couplés aux environnements d'apprentissage (Martínez et al., 2005)

Après que le format commun a été proposé, différentes options se présentent pour son utilisation dans les environnements d'apprentissage et les outils d'analyse suivant les modifications de codes sources requises. En effet, l'idéal est de modifier les codes sources des différents environnements d'apprentissage pour générer le format commun, et des outils d'analyse pour accepter le format commun en entrée. Cependant, ceci n'est pas toujours évident. Il est parfois très laborieux de modifier le code source car il peut s'avérer que cette méthode soit longue avec les risques de causer des disfonctionnements liés aux modifications. La solution adoptée est de considérer deux options selon le cas. Si la modification du code source d'un environnement d'apprentissage ou d'un outil d'analyse est simple et permet directement de générer le format commun ou de l'accepter en entrée, la solution choisie consiste à modifier le code source. Dans le cas contraire, une alternative consiste à garder le format original de l'outil et de procéder à des transformations XSLT du format des traces générées par un environnement d'apprentissage vers le format commun, ou du format commun vers le format des traces géré par un outil d'analyse. Ces différentes options d'utilisation du format commun sont illustrées dans la Figure 16 ci-dessous.

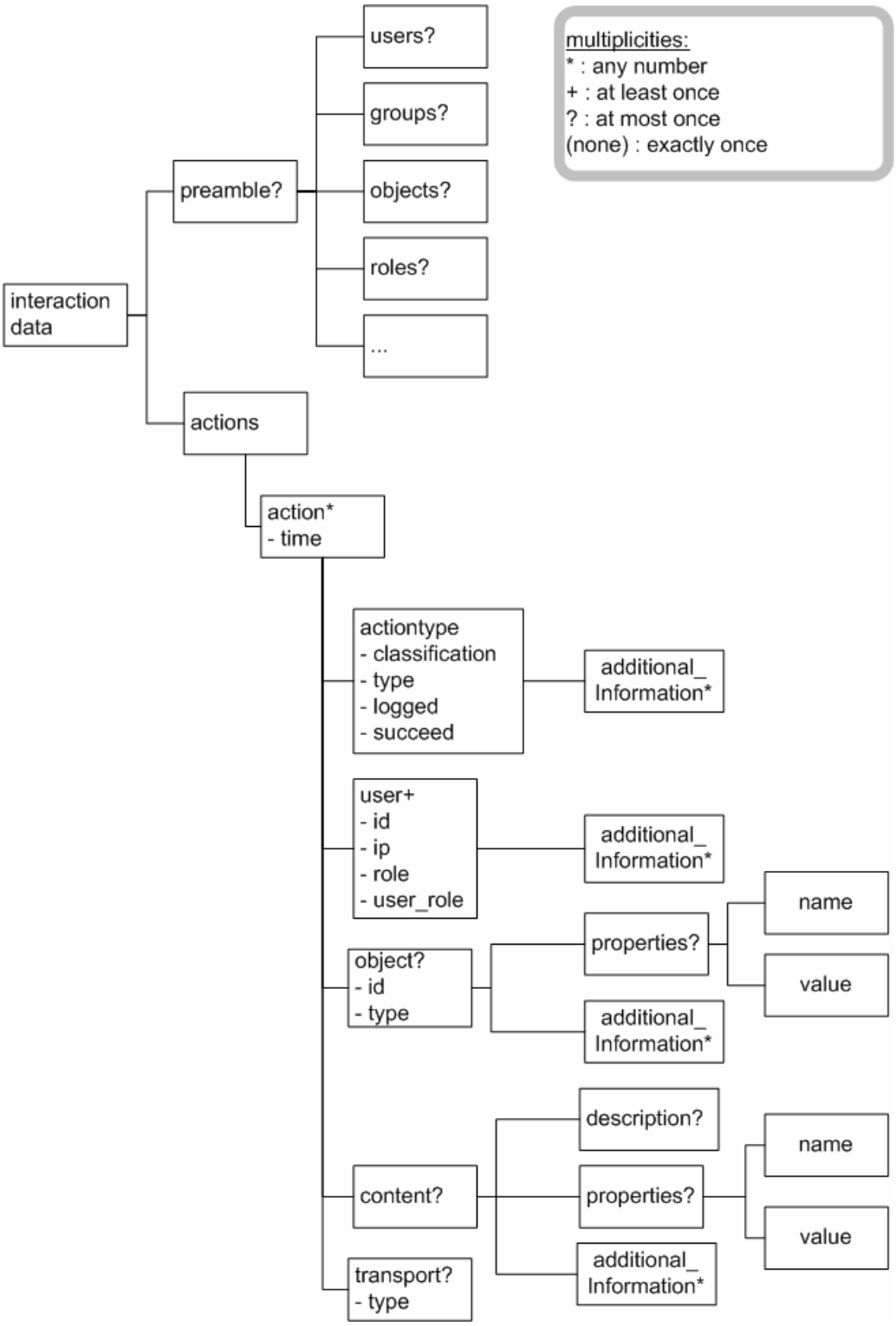


Figure 15 Partie de la DTD du format commun (Martínez et al., 2005)

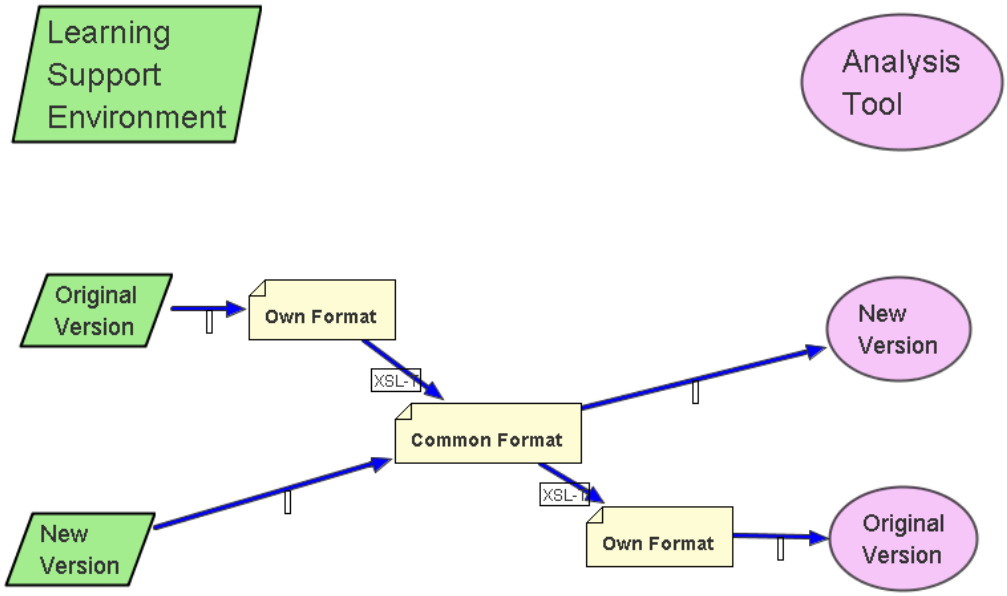


Figure 16 Différents modes d'utilisation du format commun par les environnements d'apprentissage et les outils d'analyse (Martínez et al., 2005)

L'idée de ce projet est très intéressante. En effet, il est nécessaire de partager les travaux réalisés dans différentes équipes pour promouvoir le domaine des EIAH. L'idée de proposer un format commun pour représenter les traces d'interaction et de construire une librairie d'outils d'analyse est très prometteuse. Cependant, nous n'avons pas pu évaluer les résultats de ce projet car aucune librairie utilisable des outils partagés n'a été ouverte au public.

2.2.11 Tutor Message Format, PSLC Datashop

Pittsburgh Science of Learning Center Datashop (PSLC datashop, 2013) propose un entrepôt de données des interactions entre les apprenants et les outils d'apprentissage exploitables avec un ensemble d'outils d'analyse et de reporting. Datashop est un entrepôt de données et une application Web destiné aux chercheurs travaillant dans le domaine des sciences de l'apprentissage et en particulier sur les systèmes de type « tuteur intelligent » (ITS : Intelligent Tutoring System). Un ITS est un système informatique qui a pour but de fournir un enseignement immédiat et de personnaliser les instructions et rétroactions destinées aux apprenants, généralement sans l'intervention d'un enseignant humain (Psotka et Mutter, 1988). Les situations d'apprentissage prises en considération sont individuelles. L'entrepôt offre un stockage sécurisé des données tandis que l'interface Web fournit un ensemble d'outils d'analyse et de visualisation ((PSLC datashop, 2013), (Koedinger et al., 2008)).

N'importe quel chercheur travaillant avec des ITS peut stocker ses données dans datashop afin de les partager avec d'autres chercheurs et d'utiliser les outils d'analyse et de visualisation offerts par la plateforme. Pour ceci, le chercheur doit exprimer ses données selon le format « tutor message format » (Tutor Message Format, 2013), structurées dans un document XML et respectant la DTD ou le schéma XML spécifiés.

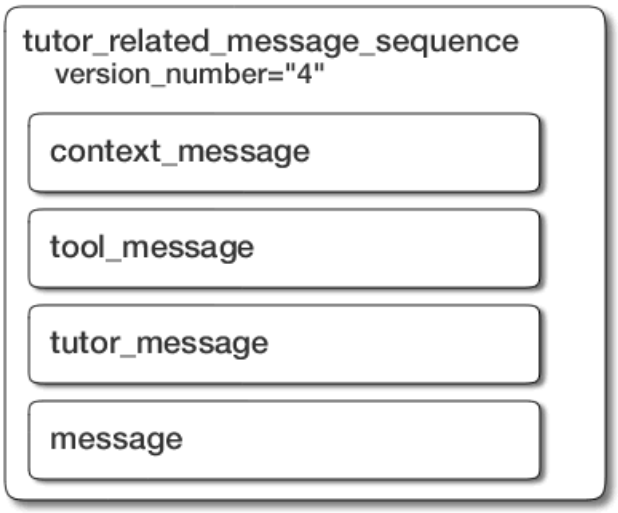


Figure 17 Types des messages tracés (Tutor Message Format, 2013)

La Figure 17 ci-dessus illustre les différents types de messages traçables par le Tutor Message Format, qui constituent une trace des interactions Homme/Machine collectées durant une session d'apprentissage réalisée par un apprenant avec l'outil tuteur intelligent. L'élément « context_message » établit le contexte de la séquence des messages qui suivent, permettant ainsi d'éviter les redondances. Cet élément permet par exemple, si l'action est réalisée dans une salle de cours, de collecter les détails correspondants (nom classe, école, période, etc.). Il permet également de préciser d'éventuelles conditions de recherche particulières sur les données, et de lier les messages tracés à un problème ou une activité appropriée. Cet élément peut aussi contenir les compétences ou composant de connaissance (knowledge component) nécessaires pour résoudre le problème ou une activité. L'élément « tool_message » décrit l'interaction d'un apprenant avec l'outil tuteur intelligent utilisé, il est aussi utilisé pour représenter une action réalisée par l'outil. L'élément « tutor_message » décrit la réponse du tuteur intelligent à l'action de l'apprenant contenu dans un élément « tool_message » (par exemple un apprenant demande de l'aide pour une étape donnée (tool_message), le tuteur intelligent lui répond alors en fournissant un indice (tutor_message)). L'élément « message » est utilisé pour représenter des informations non prises en charge par les autres éléments, par exemple CTAT (projet de recherche à Carnegie Mellon University ayant pour but de

construire des outils permettant le développement simplifié de tuteurs intelligents (CTAT, 2003), ces derniers donnent la possibilité de tracer directement vers les serveurs PSLC, et ceci par un simple paramétrage) utilise ce type de message pour représenter une action de rejouage (replay) (un étudiant qui rejoue une suite d'actions qu'il a effectué).

PSLC datashop a réussi à former une communauté importante autour des services de dépôt et d'analyse qu'il offre, favorisant le partage de données et d'outils d'analyse. Le Message Tutor Format doit être utilisé pour structurer les traces provenant des ITS et bénéficier des outils d'analyse, de visualisation et de reporting offerts par la plateforme.

2.2.12 UTL

UTL (Usage Tracking Language) (Choquet et Iksal, 2007) est un langage de modélisation de la trace, indépendant du langage de modélisation pédagogique utilisé par le concepteur pédagogique et des formats des traces générées par le dispositif d'apprentissage. Une trace est définie comme « toute donnée fournissant de l'information sur une session d'apprentissage » (Choquet et Iksal, 2007). UTL considère la trace comme un objet pédagogique au même titre qu'un scénario pédagogique ou qu'une ressource pédagogique. Afin de permettre une utilisation optimale et effective des traces pour la réingénierie d'EIAH, l'idée d'UTL consiste à donner la possibilité au concepteur pédagogique d'intervenir dans la modélisation de l'observation. Le concepteur est considéré comme étant le mieux placé pour spécifier le contenu de la trace, lui donnant ainsi la possibilité d'exprimer le comportement prévu (scénario prédictif (Lejeune et Pernin, 2004)) afin de le comparer à l'activité réellement réalisée par l'apprenant (scénario descriptif). L'objectif est d'exprimer les traces dans un niveau d'abstraction élevé pour que le concepteur les comprenne et puisse en profiter dans la réingénierie de l'EIAH.

UTL propose le modèle de trace DGU illustré dans la Figure 18 ci-dessous. C'est un modèle à trois facettes : (1) la facette « Définition » permet de modéliser le besoin d'observation en donnant la possibilité au concepteur de modéliser la trace nécessaire à ses analyses ; (2) la facette « Obtention » permet de modéliser le moyen de l'observation ; (3) et la facette « Utilisation » permet de décrire l'utilisation de ces traces.
Ce modèle donne la possibilité aux concepteurs de situations pédagogiques et aux développeurs de coopérer. En effet, le concepteur pédagogique commence par la *définition* de

ce qu'il souhaite observer, et la description de l'objectif d'*utilisation*, afin de pouvoir négocier avec le développeur des moyens d'*obtention* de la trace.

Deux versions d'UTL ont été proposées : (1) UTL1 composé de UTL/S (S pour scénario) et UTL/T (T pour trace). UTL/S permet d'exprimer les concepts traçables du scénario pédagogique dans un langage indépendant du langage de modélisation pédagogique (par exemple IMS-LD (IMS-LD, 2003)). UTL/T permet d'extraire à partir des traces générées automatiquement par un environnement d'apprentissage des traces significatives exprimées dans un langage indépendant du format original des traces (par exemple un fichier de log) ; (2) UTL2 (cf. Figure 19) rajoute à UTL/S et UTL/T la partie UTL/P (P pour patron). Cette version s'intéresse à la capitalisation des savoir-faire techniques d'analyse de l'utilisation d'un EIAH par l'expression d'indicateurs sous forme de patrons de conception.

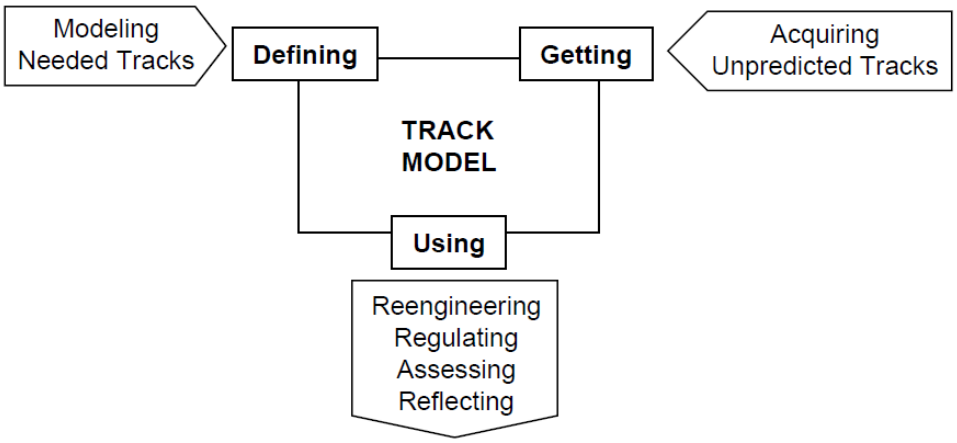


Figure 18 Le modèle DGU (Defining Getting Using / Définition Obtention Utilisation) (Choquet et Iksal, 2007a)

La Figure 19 illustre les différents types de données manipulées par UTL. Deux types de données sont distingués : les données primaires (primary-datum) et les données dérivées (derived-datum). Une donnée primaire peut être : (1) une donnée brute (raw-datum), collectée par le dispositif d'apprentissage, par exemple un fichier de log ; (2) une donnée de production (content-datum), qui représente une production d'un acteur d'une situation d'apprentissage, par exemple un travail réalisé par un apprenant ; (3) une donnée additionnelle (additional-datum), liée à une situation d'apprentissage et utilisée dans le calcul d'une donnée dérivée, par exemple une donnée ad-hoc ou une taxonomie.

Une donnée dérivée est calculée à partir de données primaires et/ou d'autres données dérivées. Une donnée dérivée est soit une donnée intermédiaire (intermediate-datum) utilisée

dans le calcul d'une autre donnée, soit un indicateur signifiant sur le plan pédagogique et calculé à partir des données observées pour évaluer la qualité de l'interaction, de l'activité et de l'apprentissage dans un EIAH. Un indicateur est lié à un contexte pédagogique qui est défini par un objectif d'observation (tracking-purpose) et par un concept traçable (traceable-concept).

La Figure 20 illustre le modèle de la trace défini par UTL/T. Ce modèle de type clé-valeur est conçu pour être générique et compatible avec la majorité des formats de traces collectées automatiquement. L'élément « category » définit l'une des catégories possibles pour les données composant une trace: les mots clés et les valeurs. L'élément « title » d'une trace associe un sens à son contenu. L'élément « data » permet de stocker le contenu du mot clé ou de la valeur. L'élément « type » est une liste ouverte des types de format de stockage de la trace initiale, il peut prendre la valeur « text », « xml », « database », etc. L'élément « path » est utilisé pour retrouver le chemin d'accès à une donnée quand la trace est stockée dans un fichier XML (chemin sous forme de requête xpath) ou dans une base de données (requête sql). Sinon, si les traces sont stockées dans un fichier de logs structuré au format texte, les données sont localisées en utilisant les éléments « begin » et « end » ou les éléments « delimiter » et « position ».

UTL propose une méthode très intéressante par sa capacité de permettre au concepteur pédagogique de participer à la modélisation de l'observation. Le concepteur pédagogique est désormais capable de comparer le scénario prédictif qui représente l'usage attendu du dispositif d'apprentissage et le scénario descriptif relatif à une session d'apprentissage effective réalisée par un apprenant. Le modèle de trace proposé est très générique et correspond plutôt à un modèle permettant d'extraire à partir des traces brutes celles qui intéressent le concepteur pédagogique dans son étude. Ce modèle ne peut donc pas être directement considéré pour tracer les interactions dans un EIAH.

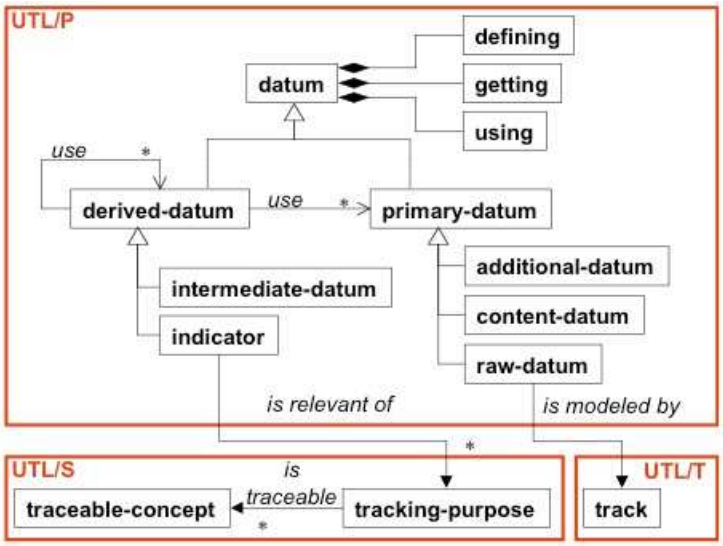


Figure 19 Modèle conceptuel du Méta-Langage UTL2 (Choquet et Iksal, 2007)

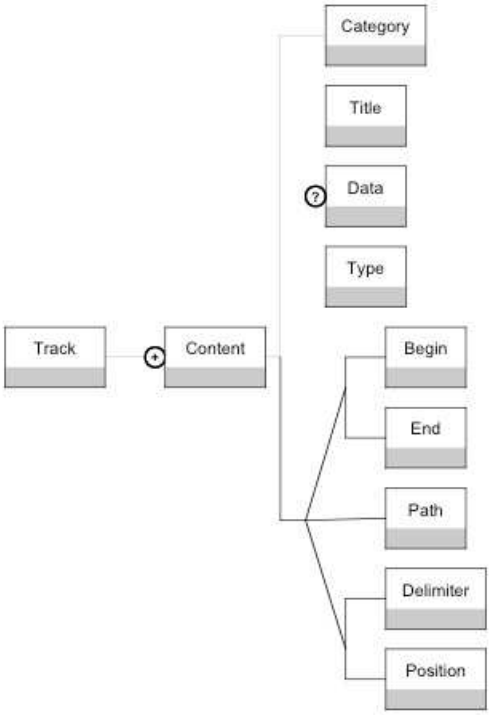


Figure 20 Le modèle d'information d'une trace (UTL/T) (Choquet et Iksal, 2007)

2.2.13 MULCE

Le projet MULCE (Multimodal Learning and teaching Corpora Exchange) (Chanier et al., 2010) se place dans le contexte du partage de données d'interaction générées par des environnements d'apprentissage collaboratifs, permettant d'effectuer des analyses différentes dans diverses disciplines. Pour cela, le projet propose une structure commune et générale de corpus de traces d'interaction. Cette structure sera présentée en détail dans la deuxième partie

de cet état de l'art. Le partage des données et des analyses permet la réplication, la vérification et la contradiction de résultats de recherche déjà établis. Ce qui nous intéresse dans cette section est la formalisation utilisée par MULCE pour la représentation des données d'interaction. Les membres du projet MULCE ont proposé un format spécifique pour structurer les données d'interaction issues de situations d'apprentissage collaboratives. La proposition de ce nouveau format est justifiée par l'absence d'un format standard permettant de représenter les traces d'interaction produites par un environnement d'apprentissage collaboratif.

La Figure 21 illustre une partie du schéma XML proposé pour décrire le format proposé des traces d'interaction. Un environnement utilisé dans une formation est décrit comme un espace de travail (workspace) relatif à un lieu offrant des outils ayant des fonctionnalités associées à des acteurs en interaction pour une période donnée. Un espace de travail peut inclure des sous-espaces de travail. Cette définition récursive permet de choisir le niveau de granularité des données d'interaction pour un corpus donné. Un espace de travail peut donc représenter la formation, une étape ou une activité, ce qui permet d'organiser les interactions selon différentes perspectives (activité, tranche temporelle, type ou espace d'interaction). Un espace de travail contient une référence vers les acteurs participant à la formation, les dates de début et de fin, les outils d'interaction disponibles, et la liste des actes (chaque acte fait référence à un outil offert par l'environnement d'apprentissage).

La Figure 22 illustre la définition générique de la notion d'acte. Un acte contient : une référence vers l'outil l'ayant produit, une référence vers l'auteur à l'origine de l'interaction, une date de début et un sélecteur de types d'actes (act_type). Ce dernier permet de spécifier le type d'acte (ex : chat, forum, mail) et d'ajouter à la définition d'un acte une partie spécifique au type d'acte en question.

La Figure 23 présente le schéma d'un acte de type forum. Un message posté dans un forum possède un sujet et peut faire référence à un message père auquel il répond. Le contenu du message contient trois types d'éléments : le message, une citation et une signature. Si le message a été lu, l'acte contient la date de lecture ainsi qu'une référence vers l'acteur concerné. Enfin, un fichier attaché peut être référencé dans un message de forum par son nom, son type, sa description et sa date.

Le schéma de trace proposé a été conçu pour être exhaustif et permettre la description des différentes fonctionnalités des divers outils de communication utilisables dans les EIAH. Ce format reste cependant extensible pour ajouter de nouveaux outils ou fonctionnalités non pris en compte permettant ainsi la description de corpus variés.

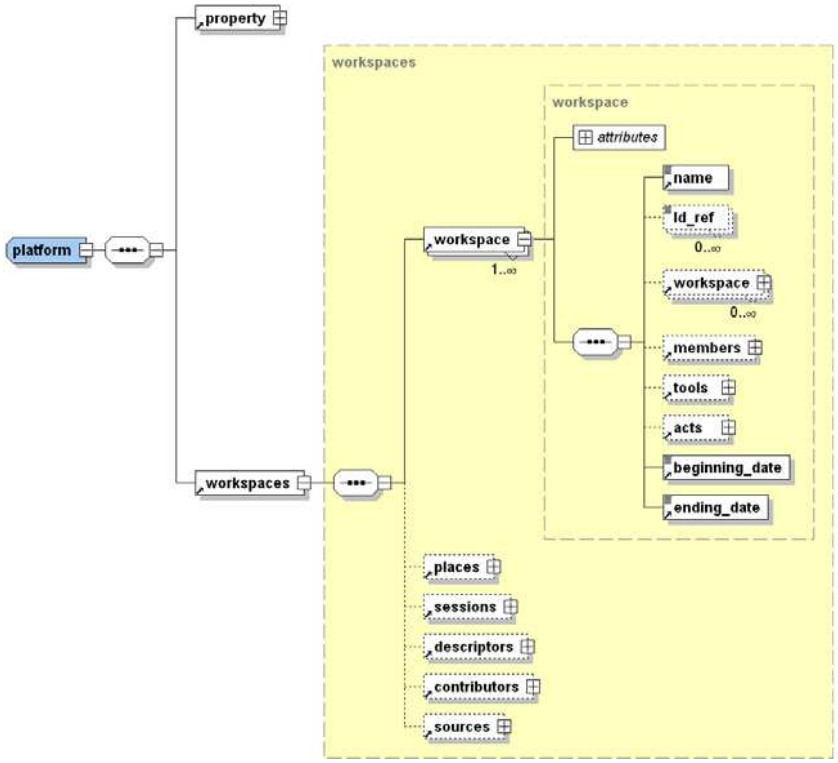


Figure 21 Extrait du schéma XML du format des traces d'interaction proposé (Reffay et al., 2008)

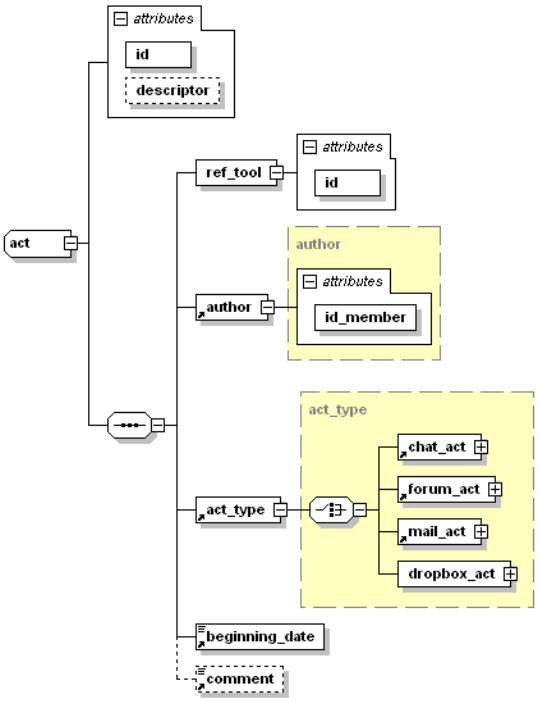


Figure 22 Extrait du schéma XML, notion d'acte (Reffay et al., 2008)

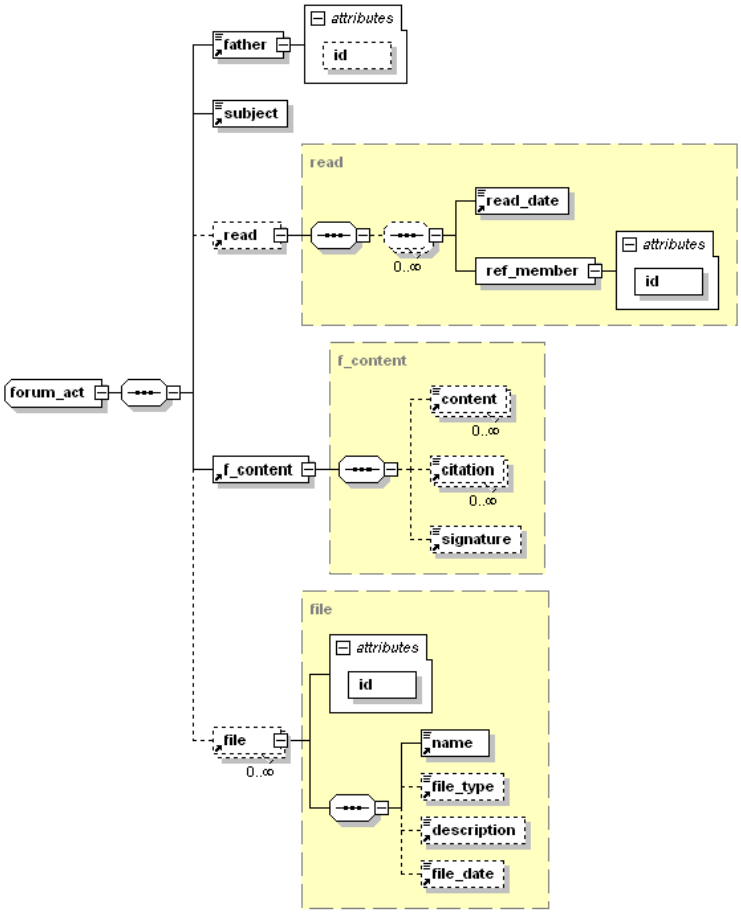


Figure 23 Extrait du schéma XML, Concept d'acte de forum (Reffay et al., 2008)

2.3 Partage de corpus de traces et d'outils d'analyse

Nous nous intéressons au concept de partage dans le domaine de l'apprentissage humain assisté par ordinateur. En effet, afin d'améliorer les EIAH, les acteurs engagés dans leur évaluation ont besoin d'analyser les usages de ces EIAH. Ces analyses requièrent la collecte des traces des interactions provenant de l'utilisation des EIAH, mais aussi l'utilisation d'outils d'analyse et de visualisation permettant d'analyser les données collectées. Des travaux ont déjà étudié les possibilités de partage dans un objectif d'évaluation, de validation, de réutilisation et de capitalisation permettant d'atteindre une utilisation plus large des environnements informatiques, dépassant ainsi l'étape des prototypes. Deux types de partage complémentaires sont considérés : le partage de données et le partage d'outils d'analyse et/ou de visualisation. Les données partagées peuvent être de deux catégories ; les données d'interaction provenant de l'utilisation des EIAH et les données provenant du processus d'analyse des données d'interaction.

Un chercheur publiant les résultats de son travail dans n'importe quel domaine scientifique et en particulier dans le domaine des EIAH devrait permettre à d'autres chercheurs de la communauté de vérifier la validité des résultats publiés, de les contredire ou de les compléter (Reffay et al., 2008). Dans (King, 2007), l'auteur s'intéresse au partage systématique des données ayant servies à des études statistiques en sciences sociales par des chercheurs qui publient des articles scientifiques décrivant leurs travaux. Il décrit une architecture d'une plateforme Web (Dataverse Network, 2013) qui affecte aux données, grâce à un algorithme de cryptographie, un identifiant unique appelé UNF (*Universe Numeric Fingerprint* ou Empreinte Digitale Universelle) permettant de vérifier que les données fournies sont les mêmes que celles décrites dans une publication ou si elles ont subies des modifications. Cette architecture permet d'accéder aux données et de leur offrir une meilleure visibilité et un meilleur taux de citation. Il est également intéressant d'archiver des données dans le but de les réutiliser ; en les analysant par une nouvelle version de l'outil d'analyse ou un nouvel outil, ces données peuvent être considérées dans ce cas comme un corpus de test de référence permettant de faire du benchmarking. Par ailleurs, le partage de données provenant d'une expérimentation réalisée par une équipe de recherche avec d'autres chercheurs provenant d'équipes différentes suppose la contextualisation (Reffay et al., 2008) des données

partagées permettant aux chercheurs qui n'ont pas participé à l'expérimentation de comprendre les contenus des corpus.

L'intérêt du partage des données est très souvent lié au partage d'outils d'analyse et/ou de visualisation de ces données. En effet, partager des données sans pouvoir les exploiter n'a d'intérêt que pour construire une archive. Par ailleurs, un outil d'analyse et/ou de visualisation disponible mais non accompagné de données au bon format d'entrée ne bénéficiera pas d'une utilisation large de la part d'équipes autres que celle l'ayant développé, d'où l'importance de la disponibilité d'un corpus servant à tester un outil partagé. Par exemple, en téléchargeant l'outil d'analyse TATIANA (Tatiana, 2009) (Dyke, 2009), celui-ci est accompagné d'un corpus permettant de tester les fonctionnalités de l'outil.
Nous présentons dans cette partie un ensemble de travaux qui se sont intéressés au partage de données et d'outils d'analyse et/ou de visualisation dans le domaine des EIAH.

2.3.1 Partage de données et d'outils

Le développement technologique ne cesse de s'accroitre à une vitesse importante dans le domaine des EIAH. Pour permettre l'aboutissement et l'utilisation des EIAH développés, il est indispensable de pouvoir partager les travaux réalisés au sein de la communauté permettant un gain de temps en évitant de reprendre des travaux déjà réalisés. Dans ce sens, beaucoup de travaux se sont intéressés au développement de méthodes et d'outils d'analyses des interactions issues de situations d'apprentissage médiatisées par ordinateur. Ces outils peuvent être utilisés dans un objectif de réingénierie d'un EIAH permettant de détecter et de traiter rapidement les disfonctionnements apparus au niveau des fonctionnalités ou d'un éventuel scénario pédagogique.

Une équipe développant son propre EIAH peut réutiliser des outils d'analyse développés par d'autres équipes pour évaluer son travail. Le partage d'outils de sources différentes suppose une interopérabilité au niveau des données échangées et analysées.

Par ailleurs, il est très intéressant pour des chercheurs d'équipes différentes de partager et réutiliser des corpus de données, issus d'expérimentations. Les données partagées peuvent être réutilisées pour vérifier l'exactitude de résultats publiés ou les contredire. Elles peuvent aussi servir pour de nouvelles analyses complémentaires ou comme base de test pour de nouvelles versions de méthodes d'analyse.

Les sections suivantes font le tour de différents projets qui se sont intéressés au partage de données et d'outils d'analyse et/ou de visualisation dans le domaine des EIAH.

2.3.2 IA JEIRP – Librairie d'outils d'analyse des interactions

Le projet IA JEIRP (Martínez et al., 2005) propose de construire une librairie d'outils d'analyse des interactions. Comme déjà présenté dans la section précédente, ce projet a pour objectif de découpler un ensemble d'outils d'analyse étroitement liés à un ensemble d'environnements d'aide à l'apprentissage. Cette idée provient de la constatation qu'un outil d'analyse développé par une équipe peut fournir des fonctionnalités utiles pour une autre équipe. Il est donc judicieux de trouver un moyen pour réutiliser un outil existant au lieu d'en développer un autre spécifique et fortement couplé à l'EIAH. La contrainte technique qui s'impose est le format des données manipulées par les outils d'analyse à partager, d'où la proposition du format commun (déjà présenté) pour représenter les traces d'interaction. Cependant, après la fin de ce projet, la plupart des outils d'analyse sont inaccessibles ou insuffisamment documentés pour être réutilisés. Le format commun a été réutilisé dans le projet CAVICOLA (Computer-based Analysis and Visualization of Collaborative Learning Activities) (Harrer et al., 2007). Ce projet propose un modèle flexible pour la construction de processus d'analyse et de visualisation des activités d'apprentissage collaboratives. Ce modèle utilise le format commun comme format de collecte des traces à analyser.

2.3.3 REDiM

Le projet REDiM (Réingénierie des EIAH Dirigée par les Modèles) (Choquet, 2007) s'intéresse à la réingénierie des scénarios pédagogiques dans les EIAH en utilisant les traces, et en particulier les fichiers de logs générés par un EIAH au cours d'une session d'apprentissage. Les traces générées par l'EIAH représentent le scénario descriptif correspondant à l'activité réalisée. Ce scénario descriptif est comparé au scénario prédictif conçu par le concepteur pédagogique (Lejeune et Pernin, 2004). Cependant, les traces générées sous forme de fichier de logs ne sont généralement pas exploitables par le concepteur pédagogique du fait de leur bas niveau d'abstraction. La solution apportée par ce projet consiste en la proposition d'une architecture ouverte d'outils d'analyse distribués (Iksal et Choquet, 2005) permettant une interprétation compréhensible par le concepteur pédagogique des traces contenus dans des fichiers de log. Le concepteur pédagogique,

pouvant être un enseignant (milieu académique) ou un formateur (milieu de l'entreprise), cette architecture permet de l'intégrer dans le processus d'ingénierie et de réingénierie de l'EIAH. L'architecture proposée (cf. Figure 24), fournit une collection d'outils d'analyse aux chercheurs ou concepteurs souhaitant analyser l'usage d'un EIAH. L'approche choisie est proche des services Web. L'architecture proposée offre un ensemble de serveurs autour d'un serveur spécial appelé « routeur ». Chaque serveur offre un ensemble de services et doit s'inscrire auprès du « routeur ». Un service est composé d'un ensemble de méthodes lié à un concept ou un domaine spécifique. Un exemple de service peut être l'importation de fichiers de logs, et ce service propose alors une méthode par format de log. L'architecture proposée donne la possibilité à un chercheur dans l'analyse des usages d'ajouter et de partager de nouveaux services, qui peuvent éventuellement combiner les sorties d'autres services. Toutes les méthodes sont accessibles à partir de n'importe quel serveur. Cette architecture est ouverte et distribuée, il est donc possible de connecter de nouveaux serveurs à l'ensemble des serveurs existants. Les services sont accessibles depuis n'importe quel serveur, il n'est donc pas nécessaire, pour utiliser un service, de connaître le serveur qui l'héberge. Un analyste voulant accéder à une méthode n'a pas besoin de connaître le serveur qui l'offre mais seulement la signature de cette méthode c'est-à-dire son nom, les entrées et les sorties.

Pour permettre l'interopérabilité entre les services offerts, un langage de représentation des traces d'usage a été proposé. Baptisé UTL (Usage Tracking Language), ce langage est utilisé pour exprimer les traces à analyser par les services offerts et sert de langage intermédiaire entre le langage de modélisation pédagogique utilisé par le concepteur pédagogique pour modéliser le scénario pédagogique d'une part, et le format des fichiers de logs générés par les environnements d'apprentissage.

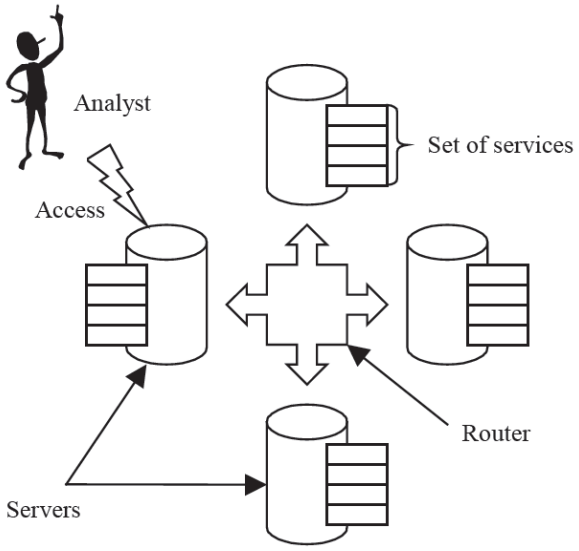


Figure 24 Architecture d'outils d'analyse distribués (Iksal et Choquet, 2005)

Ce travail a pour objectif l'intégration du concepteur pédagogique dans le processus de l'analyse des interactions lui permettant d'analyser les traces d'interaction en les élevant à un niveau d'abstraction qui lui est compréhensible. Pour ceci, une architecture de partage de services destinés à l'analyse des traces d'interaction a été proposée. Cette architecture est centrée sur le langage UTL pour la représentation des traces indépendamment de leur origine. Cette proposition est très intéressante et peut être d'une grande utilité pour la communauté EIAH. Cependant, en pratique, seuls deux exemples de services ont été présentés dans (Iksal et Choquet, 2005) et il n'y a pas encore eu de diffusion d'une plateforme permettant de tester des services existants ou d'en intégrer des nouveaux.

2.3.4 PSLC Datashop

Pittsburgh Science of Learning Center (PSLC) Datashop (PSLC datashop 2013) est un projet qui contribue au domaine de l'« Educational Datamining ». L'Educational Datamining[8] est une discipline émergente, concernée par le développement de méthodes permettant d'explorer les types de données qui proviennent des dispositifs éducatifs, et l'utilisation de ces méthodes afin d'étudier les apprenants et les processus par lesquels ils apprennent. PSLC Datashop offre deux types de services à la communauté EIAH, et en particulier pour les systèmes de type « tuteur intelligent ». En effet, Datashop met à la disposition des chercheurs : (1) un service de dépôt sécurisé des données d'interaction générées dans un ITS (Intelligent Tutor System, tuteur intelligent) au cours d'une session d'apprentissage, Datashop

[8] Définition traduite de l'anglais disponible sur http://www.educationaldatamining.org/

favorise ainsi le partage de données de recherche ; (2) un ensemble d'outils d'analyse et de reporting partagés permettant d'analyser les données d'interaction.
Afin de pouvoir bénéficier des outils d'analyse offerts par Datashop, un chercheur doit exprimer les données d'interaction à analyser selon le format « Tutor Message Format » (Tutor Message Format, 2013). Aux données d'interaction, un chercheur peut joindre des articles liés ainsi que des fichiers contenant plus de descriptions et d'informations concernant les expérimentations en question.

Par ailleurs, Datashop offre la possibilité d'exporter les données d'interaction délimitées par des tabulations sous forme de fichier texte, il est donc possible de les analyser avec des outils d'analyse externes. À l'importation de données d'interaction dans Datashop, celles-ci sont accompagnées d'un ou de plusieurs modèles de composant de connaissance (Knowledge Component Model). Un composant de connaissance est considéré comme toute connaissance nécessaire à la résolution d'une étape d'un problème. Un modèle de composant de connaissance fournit une liste de correspondances entre chacune des étapes de résolution d'un problème et un ou plusieurs composants de connaissance.

Un chercheur qui importe ses données dans Datashop a le choix de les offrir avec un accès public ou privé. Il est ainsi possible pour un chercheur de tester les outils d'analyse proposés par Datashop sur des données publiques, et de décider par la suite de la pertinence de ces outils pour analyser ses propres données.

Un exemple d'outil d'analyse est donné à la Figure 25, qui illustre deux courbes d'apprentissage selon deux modèles de compétences différents. La courbe représente le taux d'assistance (nombre d'indices et d'essais incorrects) en fonction de l'opportunité (chance pour un apprenant de démontrer s'il a appris un composant de connaissance donné).

Ce projet propose des outils intéressants pour les chercheurs utilisant des ITS. L'originalité de ce projet réside dans le fait de proposer un cadre complet permettant aux chercheurs de tracer leurs données et de les stocker en temps réel ou différé dans la plateforme Datashop, d'analyser ces traces, et de comparer leurs résultats à ceux des autres chercheurs. Cependant, les chercheurs ayant des données exprimées dans des formats différents du « Tutor Message Format » sont contraints de convertir leurs données pour pouvoir profiter des outils offerts. Par ailleurs, ce projet s'intéresse uniquement aux ITS et ne traite pas d'autres types d'EIAH.

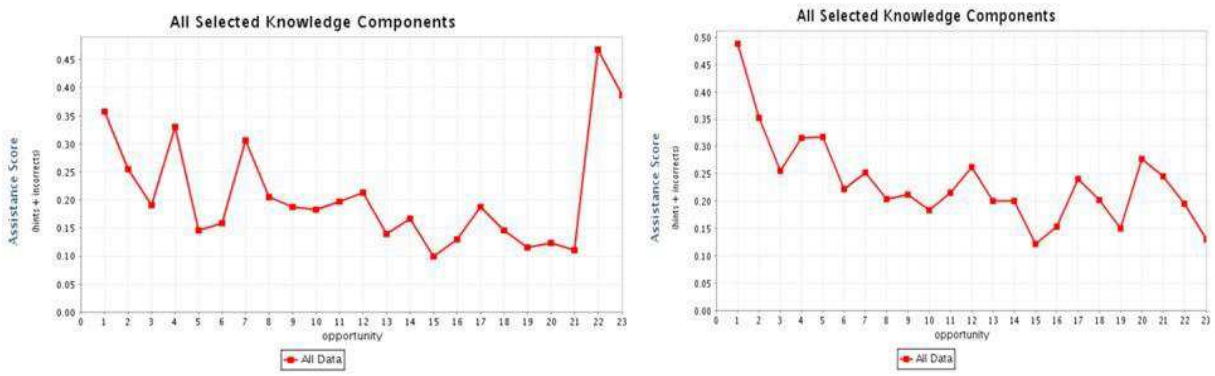


Figure 25 Courbes d'apprentissage avec différents modèles de compétence (PSLC Datashop, 2013)

2.3.5 MULCE

Le projet MULCE ((Chanier et al. 2010), (Reffay et al., 2008), (Reffay et Betbeder., 2009)) s'intéresse au partage de corpus de données d'interaction issues de situations d'apprentissage en ligne et des outils d'analyse associés à ces corpus. Le besoin derrière ce partage provient du fait que des chercheurs publient des résultats scientifiques qui ne peuvent pas être vérifiés, validés ou contredits par d'autres chercheurs. Par ailleurs, de nombreux travaux ne dépassent pas le niveau du prototype et ceci est dû au manque de validation et de partage au sein de la communauté EIAH. L'idée du projet MULCE est de permettre le partage de données d'interaction collectées dans des expérimentations utilisant des EIAH afin de donner accès à d'autres chercheurs pour vérifier la validité des résultats scientifiques publiés, ou encore pour construire de nouvelles analyses ou enrichir des analyses antérieures. Afin d'atteindre ces objectifs, les traces d'interaction peuvent ne pas suffire. En effet, un chercheur n'ayant pas participé à l'expérimentation étudiée et ne connaissant pas son contexte ne pourra pas facilement comprendre les traces et les analyser. Dans (Noras et al., 2007), un corpus de formation en ligne est défini comme : « un ensemble de données et de traces issues d'une expérimentation, enrichies par des informations techniques, humaines, pédagogiques et scientifiques permettant leur analyse en contexte. »

MULCE propose une structuration de corpus d'apprentissage et d'enseignement (LETEC : Learning and Teaching Corpora) permettant l'expression et le partage de données hétérogènes d'interaction dans leur contexte. La Figure 26 illustre les différents constituants d'un corpus LETEC. La structure d'un corpus LETEC proposée par le projet MULCE comprend quatre composants : (1) le *contexte* est composé à son tour de deux sous-composants : (i) le *scénario pédagogique* permettant d'exprimer le déroulement attendu d'une situation d'apprentissage. Il définit les objectifs pédagogiques, les pré-requis et les contenus

des activités et tâches nécessaires au déroulement de l'apprentissage. Il permet également de décrire les rôles des différents membres participants. (ii) Le *protocole de recherche*, constituant facultatif, permet d'exprimer les questions de recherche qui sous-tendent l'expérimentation. Il permet de décrire les rôles des acteurs impliqués, ainsi qu'un ensemble d'activités planifiées avant, après ou au cours de la réalisation de l'expérimentation. Par exemple prévoir des activités de remplissage de questionnaires de recherche ou des entretiens avec les participants ; (2) le noyau du corpus, appelé *instanciation*, représente l'objet d'étude. L'instanciation regroupe l'ensemble des données d'interaction et des productions des acteurs de la situation d'apprentissage ainsi que les traces système ; (3) le composant *licence*, dans sa partie publique, permet de définir les droits de l'éditeur et des utilisateurs du corpus ainsi que les éléments de respect de l'éthique pour la protection de la vie privée des acteurs de la formation. Une partie de ces données reste privée, et seul le responsable du corpus peut y accéder, elle contient les patronymes et coordonnées des participants et leur accord de consentement éclairés signés. La définition de licence d'utilisation favorise le partage des données recueillies et la construction d'analyses sur ces données ; (4) Le travail de collecte et de structuration du corpus a comme objectif l'exploitation des données collectées à des fins d'analyse. Le composant *analyses* contient les analyses faites sur les données d'interaction dans un objectif de partage et de capitalisation d'analyses de différents niveaux. La transcription d'une vidéo est considérée comme un exemple d'analyse. Une transcription ou toute autre analyse est liée aux données de la partie instanciation ou à d'autres analyses.

Deux types de corpus ont été proposés par le projet MULCE. Le premier type est le « corpus global » ayant la structure déjà présentée et qui contient toutes les données d'interaction issues d'une situation d'apprentissage complétées par les données du contexte, les licences d'utilisation et d'éventuelles analyses telles que les transcriptions. Ce méga-corpus est très volumineux et contient un grand nombre de données qui n'intéressent généralement qu'en partie les chercheurs. MULCE utilise alors deux types de corpus : le corpus global et le corpus distinguable. Un corpus global d'apprentissage est un corpus d'étude qui peut être décomposé en un ensemble de corpus distinguables « chacun correspondant au grain habituellement retenu par un chercheur pour y accomplir une analyse sur un phénomène précis ». Par exemple un corpus global contient toutes les interactions ayant eu lieu pendant une formation d'apprentissage en ligne (ex : messages échangés dans des forums, clavardage, interactions audio, etc.), alors qu'un corpus distinguable peut contenir uniquement l'ensemble des interactions dans les forums utilisés par la formation.

Trois types de corpus distinguables sont proposés par le projet MULCE en fonction des objectifs qui leur sont associés (Chanier et al., 2010). Un corpus distinguable permet de : (1) *associer publication scientifique et données* permettant de répliquer les résultats d'analyses publiés ; (2) *rassembler des données prêtes à l'analyse avec la mise en forme pour des outils/logiciels libres* (exemple de la collaboration entre le projet MULCE et le projet CALICO (Giguet et al., 2009)) ; (3) *partager des analyses avec des outils associés* (exemple de la collaboration entre le projet MULCE et le projet TATIANA ((Dyke, 2009), (Dyke et al., 2008))).

Le projet MULCE constitue un travail important de structuration pour le partage de corpus d'apprentissage et d'enseignement médiatisés par ordinateur. Il propose un cadre qui prend en considération l'hétérogénéité et la multi-modalité engendrées par ce genre de situations. Cependant, la structure proposée par le projet MULCE pour représenter les données d'interaction est orientée vers les activités d'apprentissage collaboratives, et il ne serait donc pas garanti qu'elle permette, sans extension, la représentation de traces de situations d'apprentissage individuelles.

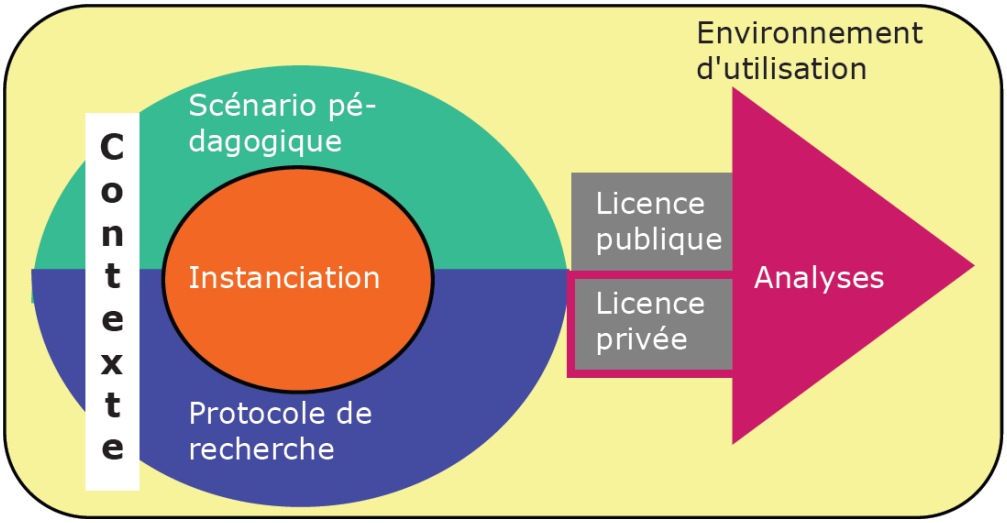


Figure 26 Constituants d'un corpus MULCE (Reffay et al., 2008)

L'intérêt pour la construction et la proposition de modèle pour la construction de corpus au sein de la communauté EIAH est récent. En effet, un chercheur travaillant sur l'étude des EIAH (ingénierie, réingénierie, analyse) effectue des expérimentations dans le cadre de ses recherches et construit un corpus de données à analyser. Cependant, il ne s'intéresse pas à la modélisation de ce corpus ni à son éventuelle réutilisation par un autre chercheur n'ayant pas participé à l'expérimentation. Le projet MULCE présente un travail intéressant dans ce sens,

il offre une banque de corpus (MULCE, 2013) mettant en ligne des corpus et un site de documentation (MULCE-Doc, 2013) contenant des ressources concernant le projet. Cependant, il manque à ce travail le développement d'outils permettant à d'autres chercheurs de déposer leurs corpus.

2.3.6 CALICO

Le projet CALICO (Communauté d'Apprentissage en Ligne, Instrumentation, Collaboration) (Giguet et al., 2009) s'intéresse au partage et à l'analyse des traces d'utilisation de forums de discussion collectées dans le cadre de formations à caractère professionnalisant se déroulant dans des situations hybrides (totalement ou partiellement à distance). Une plateforme (CALICO, 2010) permettant le partage et l'analyse de corpus de traces de forums a été développée. Une structuration a été proposée par le projet pour représenter les traces d'interaction provenant de forums de discussion. Les données collectées doivent donc être structurées selon ce format de traces, ce qui permet de les analyser en utilisant les outils d'analyse et de visualisation partagés dans la plateforme. Les outils proposés sont au nombre de sept et permettent une exploitation différente des forums suivant les besoins d'analyse. Nous présentons trois exemples d'outils (cf. Figure 27). L'outil « ShowForum » permet d'afficher les contenus des messages échangés dans un forum suivant les critères de dates, auteurs, ou fils de discussions. L'outil « Volagora » permet de visualiser des informations volumétriques liées à l'activité de discussion suivant différentes échelles temporelle (jour, semaine, mois, trimestre, semestre, année). L'outil « Concordagora » permet de rechercher les occurrences d'un terme dans les messages de forum et de les surligner, tout en affichant ses contextes droit et gauche.

Figure 27 Exemples d'outils offerts par la plateforme CALICO

L'utilisation étendue des forums de discussion a permis la proposition d'un format de représentation générique des traces de communication dans un forum. Ceci est lié au fait que les traces d'utilisation de forums contiennent plus ou moins les mêmes données. Un chercheur souhaitant utiliser la plateforme CALICO commence par structurer ses traces suivant le format de traces proposé, ce qui lui permet d'utiliser les outils d'analyse et de visualisation disponibles pour exploiter ses données. Il est également possible d'accéder à des données partagées par d'autres chercheurs. Ce projet ne fournit pas de données contextuelles permettant aux chercheurs de se renseigner sur les expérimentations ayant produit les données partagées.

2.3.7 UnderTracks

Ce projet (Bouhineau et al., 2013a) (Bouhineau et al., 2013b) a comme objectifs la conception et le développement d'une plateforme ouverte permettant la collecte, le stockage et le partage de données expérimentales provenant de l'interaction avec les environnements informatiques d'apprentissage, ainsi que la construction, le stockage et le partage des

processus d'analyse exécutés sur ces traces. Une telle plateforme permet la capitalisation de données expérimentales produites par des chercheurs dans le domaine des EIAH. Comme le projet MULCE, ce projet souligne la nécessité du partage de données provenant de situations expérimentales authentiques, vu la complexité de monter une expérimentation ainsi que l'importance de fournir les moyens nécessaires pour permettre la reproductibilité des résultats de recherche publiés.

La Figure 28 illustre le cycle de vie des données considérées par la plateforme UnderTracks. Il se compose de trois parties principales : la production des données, le traitement, et la communication. La production des données est composée de deux phases : la préparation et la collecte. Le traitement est composé de quatre phases : la validation, l'enrichissement, l'analyse, et la synthèse. L'archivage est transversal à ces différentes phases, et permet de stocker toutes les données utiles pour le partage des données. Le processus proposé est itératif et permet de revenir à la phase antérieure à partir de la phase en cours. La partie production concerne la collecte des données relatives à une expérimentation, tandis que la partie traitement concerne les processus d'analyse exécutés par les chercheurs sur les données collectées.

Les données collectées se composent des traces accompagnées de données contextuelles sous forme de méta-données décrivant l'expérimentation (par exemple : qui, quand, où, etc.), et de données relatives aux droits d'accès. Une API JavaScript est mise à la disposition des développeurs pour permettre une collecte en ligne des traces et leur stockage automatique dans l'entrepôt des traces.

Pour permettre le partage, ce projet propose une modélisation minimale de la trace d'activité d'un EIAH. Le modèle proposé est une séquence de lignes temporisées, chacune représentant une action. Les actions sont représentées par un même format composé d'éléments de base : date et heure de l'action, informations sur l'utilisateur, informations sur l'action, et informations contextuelles (pédagogiques et autres).

Concernant les processus d'analyse, la notion de brique est proposée pour permettre aux chercheurs utilisant la plateforme de composer leur processus d'analyse en réutilisant des briques existantes ou en construisant de nouvelles briques. Les processus d'analyse construits doivent être capitalisés et partagés.

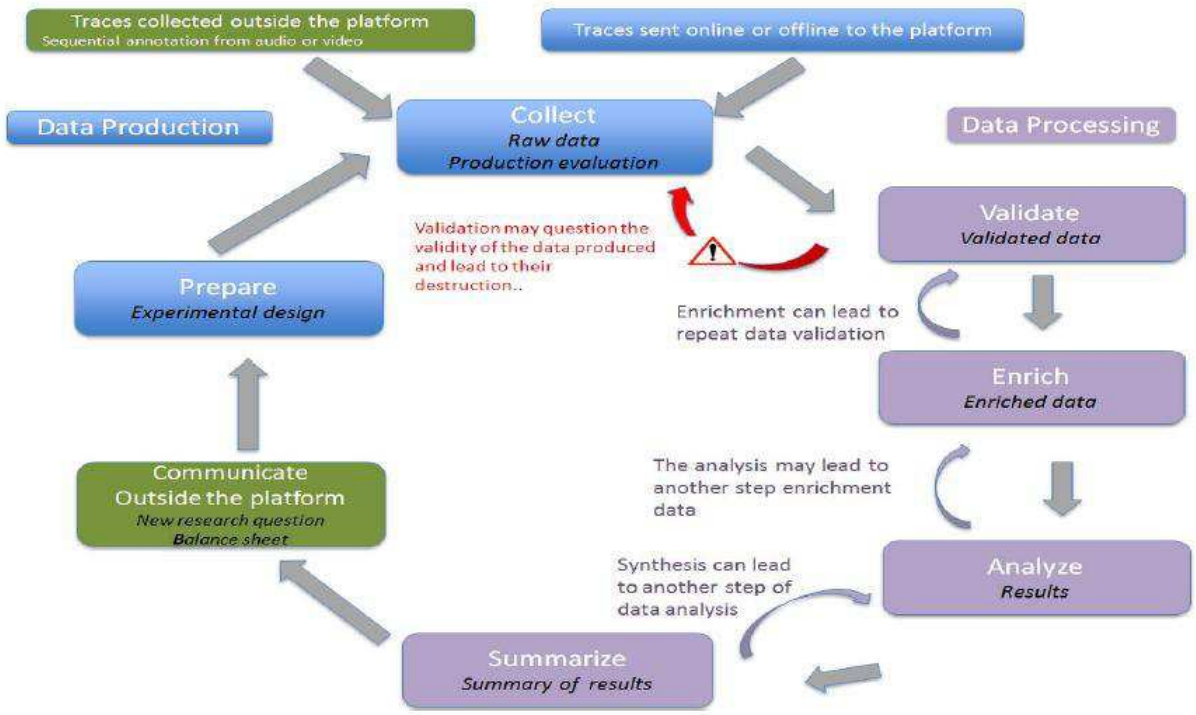


Figure 28 Le cycle de vie des données dans le projet UnderTracks (Bouhineau et al., 2013b)

Ce projet propose une approche intéressante permettant la construction et le partage de données d'expérimentation en EIAH ainsi que des processus d'analyse d'une manière intégrée dans la même plateforme. Si le développeur prévoit l'utilisation de l'Api de traçage proposée dans le développement de l'environnement informatique d'apprentissage, la collecte des données se fait d'une manière automatique. Cependant, pour partager des traces collectées dans un autre format, l'utilisation de cette plateforme nécessite un travail préalable de conversion des données collectées.

2.3.8 dataTEL

Le projet dataTel (Drachsler et al., 2010) adresse le problème de manque de corpus de données standardisés utilisables dans le développement de systèmes de recommandation dans le domaine des EIAH. De tels corpus peuvent être utilisés comme des corpus de référence servant de benchmarks pour la comparaison de différentes approches de recommandation. Les systèmes de recommandation sont développés dans le cadre de la personnalisation des EIAH pour permettre l'adaptation de l'environnement d'apprentissage aux profils des apprenants, afin d'offrir des environnements d'apprentissage plus efficaces, accélérer le développement des compétences, et développer la collaboration entre apprenants. Ces systèmes de recommandation permettent de proposer à un apprenant les personnes pouvant l'aider dans le

cadre d'une activité collaborative, et l'assistent dans la recherche d'objets et d'informations pertinentes pour son activité d'apprentissage. Le projet dataTel insiste sur l'importance du partage de corpus dans la vérification, la reproductibilité, et la comparaison des résultats de recherche publiés. Pour permettre le partage de corpus, le projet propose de représenter les données tracées dans un format commun et identifie des données minimales qui doivent être tracées : un identifiant de l'utilisateur, un identifiant de l'élément concerné par la recommandation, et le contenu ou la valeur associés à l'élément. Cette modélisation de la trace, bien qu'utilisée dans un contexte d'apprentissage, n'est pas spécifique à ce domaine. Par ailleurs, cette modélisation est fortement liée à l'application, à savoir les systèmes de recommandation.

2.3.9 Synthèse des travaux existants

L'étude des questions de recherche énoncées nous a permis d'identifier dix critères nous permettant permettant d'évaluer les travaux existants s'intéressant au partage et à l'analyse de corpus de traces. traces. Notons que les critères identifiés sont liés aux objectifs que nous nous sommes fixés pour étudier pour étudier nos questions de recherche et ne représentent pas une liste exhaustive des critères d'évaluation des systèmes de partage et d'analyse de corpus de traces d'interaction. Le tableau illustré tableau illustré dans la

Figure 29 ci-dessous récapitule l'évaluation des différents projets étudiés. Les critères considérés sont les suivants :

- Hétérogénéité des traces : les traces collectées proviennent d'environnements d'apprentissage différents et correspondent à des activités différentes ;
- Préservation de la richesse des traces initiales collectées : les données partagées préservent la richesse des traces initiales collectées qui ne sont pas altérées par les conversions ;
- Extensibilité du format de représentation des traces : possibilité d'exprimer des données non prévues dans la formalisation initiale des traces définie par le projet ;
- Contextualisation des traces : collecte de données contextuelles permettant de documenter le contenu d'un corpus et permettant à des chercheurs n'ayant pas participé à l'expérimentation de comprendre le contenu d'un corpus ;

- Apprentissage individuel : le projet traite de situations d'apprentissage individuelles. En général, les travaux qui se sont intéressé à l'apprentissage individuel étudient la modélisation des connaissances manipulées ;

- Apprentissage collectif : le projet traite de situations d'apprentissage collectives. Les systèmes qui s'intéressent à l'apprentissage collectif étudient généralement les interactions entre différents participants à l'apprentissage ;

- Outil d'analyse générique : des outils d'analyse génériques permettant l'exploitation de différents corpus sont offerts ;

- Outil d'analyse spécifique : des outils d'analyse spécifiques (liés au domaine d'application et/ou à la représentation des traces) permettant l'exploitation des corpus sont offerts ;

- Partage des analyses : les analyses réalisées sur les corpus sont partagées ;

- Reproductibilité des analyses : les analyses déjà réalisées sur un corpus peuvent être reproduites par un autre chercheur ayant accès aux données initiales.

Parmi les critères identifiés, certains peuvent être liés à deux des questions de recherche étudiées. Ainsi, les critères « hétérogénéité des traces », « préservation de la richesse des traces initiales collectées », « extensibilité des formats de représentation des traces », « contextualisation des traces », et « type d'apprentissage » sont liés à la première question de recherche qui s'intéresse à la modélisation d'un corpus de traces contextualisées. Les critères « extensibilité des formats de représentation des traces » et « type d'outil d'analyse » sont liés à la deuxième question de recherche qui s'intéresse à l'interopérabilité entre les corpus partagés et les outils d'analyse. En effet, il nous parait essentiel, pour assurer une telle interopérabilité, de permettre la représentation de traces ayant des structures et des sémantiques différentes et de permettre une interopérabilité entre ces données et des outils d'analyse spécifiques et/ou génériques. Enfin, les critères « partage des analyses » et « reproductibilité des analyses » sont liés à la troisième question de recherche qui s'intéresse à la représentation et l'intégration des analyses en vue de leur réutilisation.

En utilisant ces critères dans l'évaluation des projets étudiés, nous remarquons que suivant les objectifs de chacun des projets étudiés, seul un sous-ensemble des critères

identifiés est vérifié. Nous remarquons que les projets IA JEIR, PSLC Datashop, REDiM, MULCE, Undertracks, et CAM-CIM prennent en considération l'hétérogénéité des traces mais ne s'intéressent pas (sauf pour REDiM) à la préservation de la richesse des traces initiales collectées. En effet, la conversion des traces vers les formats définis par chaque projet peut mener à des pertes sémantiques dues au fait que la conversion permet uniquement de récupérer les données définies dans le format de traces cible. L'objectif du partage étant entre autres d'envisager des usages variées et pas forcément prévus à l'avance, nous considérons que le critère de la préservation de la richesse des traces est très important. Par ailleurs, les projets IA JEIRP, PSLC Datashop, MULCE, Undertracks, et CAM-CIM prévoient la possibilité d'extension du format de représentation des traces permettant ainsi une certaine souplesse pour gérer des traces hétérogènes. Le projet MULCE s'est particulièrement intéressé à la contextualisation des données partagées en définissant des composants contextuels dans la structure LETEC de corpus proposée. Les projets PSLC Datashop, Undertracks, et dataTel se sont également intéressé, dans une moindre mesure, à l'aspect contextualisation et son importance pour le partage de corpus. Concernant le type d'apprentissage (individuel ou collectif), nous remarquons que les projets s'intéressent généralement à l'un ou l'autre des types. En effet, l'objet d'étude porte généralement soit sur la modélisation des connaissances, soit sur l'étude des interactions. Concernant le partage d'outils d'analyse, les projets CALICO et Undertracks se sont intéressés au partage d'outils génériques alors que les projets IA JEIRP et PSLC Datashop s'intéressent aux outils d'analyse spécifiques. Le projet Undertracks, quant à lui, considère la possibilité de partage d'outils spécifiques et génériques. Seuls les projets PSLC Datashop, MULCE, et Undertracks traitent explicitement du partage des analyses réalisées sur les corpus. Enfin, les projets PSLC Datashop, CALICO, MULCE, et Undertracks permettent d'envisager la reproduction d'analyses antérieures en ayant accès aux données à analyser et les outils d'analyse correspondants.

Nous constatons que les systèmes existants ne couvrent que partiellement les critères d'évaluation identifiés. Nous présentons dans les chapitres suivants l'approche « Proxyma » que nous proposons et démontrons qu'elle permet de répondre d'une manière bien plus complète à ces critères.

Critère / Projet	Hétérogénéité des traces	Préservation de la richesse des traces initiales collectées	Extensibilité du format de représentation des traces	Contextualisation des traces	Type apprentissage		Outil d'analyse		Partage des analyses	Reproductibilité des analyses
					Individuel	collectif	générique	spécifique		
IA JEIRP	+	-	+	-	-	+	-	+	-	-
PSLC Datashop	+	-	+	+	+	-	-	+	+	+
REDiM	+	+	-	-	+	+	-	-	-	-
CALICO	-	-	-	-	-	+	+	-	-	+
MULCE	+	-	+	++	-	+	-	-	+	+
Undertracks	+	-	+	+	+	-	+	+	+	+
dataTel	-	-	-	+	+	-	+	-	-	-
CAM-CIM	+	-	+	-	+	-	-	-	-	-

Figure 29 Tableau récapitulatif de l'évaluation des travaux étudiés et qui traitent du partage de corpus et de leur analyse

2.4 Conclusions

L'étude de l'état de l'art présentée dans ce chapitre nous montre l'importance de la question de modélisation des traces dans le domaine des EIAH, considérées comme une source d'information indispensable sur le déroulement d'une session d'apprentissage médiatisée. Il n'existe pas de format standard de représentation des traces d'interaction d'apprentissage. En effet, chaque projet définit le modèle de traces qui lui convient pour collecter les données nécessaires aux phénomènes observés. Cependant, de nombreux projets ont tenté de proposer des modèles génériques permettant de structurer des traces provenant d'environnements d'apprentissage différents, tels que les projets CARTE, CAM, et IA. Les problèmes de flexibilité, d'expressivité et d'utilisabilité sont souvent soulevés. En effet, en

cherchant à être générique, les modèles gagnent à être plus flexibles et expressifs mais perdent souvent de leur utilisabilité car l'automatisation de certaines analyses devient plus compliquée. Nous avons remarqué que les chercheurs sont partagés quant au choix d'un modèle générique ou d'un modèle spécifique à l'application.

Concernant les travaux s'intéressant au partage de corpus de traces et d'outils d'analyse, la remarque principale porte sur le petit nombre de travaux qui se sont intéressés à cette problématique. Le projet MULCE, en particulier, a fait du partage de corpus de données d'interaction sa problématique principale. Son objectif étant de permettre aux chercheurs de répliquer les résultats publiés, ce projet s'intéresse essentiellement à la structuration d'un corpus et des données qui y sont partagées tout en abordant les problématiques liées à la protection de la vie privée. Nous partageons avec le projet MULCE l'idée de la nécessité de contextualiser les traces d'interactions partagées, et la nécessité d'avoir une documentation des corpus. Par ailleurs, nous présentons des propositions permettant d'offrir une infrastructure aux chercheurs afin d'utiliser et de partager des outils d'analyse. Les différents projets étudiés ont un point de base commun. Pour rendre le partage possible, ces projets commencent par proposer une modélisation des données partagées. Cette modélisation est utilisée pour formater les données partagées et pour servir d'entrée pour les outils d'analyse. En étudiant cette problématique, nous avons entamé un premier travail de modélisation (Chebil et al., 2011) pour proposer un format des traces collectées et des analyses réalisées sur ces traces, mais nous nous sommes rendu compte que nous reprenons une partie du travail réalisé par des projets précédents. Nous pensons aussi que le fait qu'un chercheur soit contraint à formaliser toutes ses données, suivant un format particulier, pour pouvoir les partager et profiter des outils d'analyse partagés peut être un inconvénient pour l'utilisabilité d'une plateforme de partage. Nous avons donc pensé à proposer une approche différente qui soit plus utilisable et flexible et qui réponde de manière plus complète aux critères d'évaluation identifiés. Cette approche est incrémentale et participative et permet de mettre en place des conversions partielles qui se concentrent sur les besoins d'analyse du chercheur. Cette approche permet de retarder autant que possible le travail de conversion qui peut paraître contraignant pour le chercheur et de préserver la richesse sémantique des données collectées. Notre nouvelle approche baptisée « Proxyma » est présentée dans la suite de ce manuscrit.

Deuxième partie
Contributions

Chapitre 3 : Verrous et propositions

3.1 Introduction

Dans ce chapitre, nous présentons notre positionnement par rapport à la problématique définie dans la première partie. Nous commençons par la définition de la notion de trace considérée dans notre travail. Nous présentons par la suite les contraintes du partage que nous avons identifiées. L'étude de ces contraintes nous a permis de proposer un modèle permettant le partage de corpus de traces contextualisées en vue de les analyser. Nous présentons ensuite des exemples de corpus construits par des équipes du projet EIAH. Enfin, nous clôturons ce chapitre par l'introduction de l'approche proposée dans ce travail.

3.2 Définition de la trace

L'un des objectifs de ce travail étant en premier lieu de partager des corpus de traces d'interaction d'apprentissage entre chercheurs, nous commençons par la définition de ce que nous considérons comme « trace ».

Nous reprenons la définition proposée dans (Lund et Mille, 2009) d'une trace numérique d'interaction comme : « une suite temporellement située d'observés, qui relève soit d'une interaction entre humains, médiatisée et médiée de diverses façons par ordinateur soit d'une suite d'actions et réactions entre un humain et un ordinateur. Cette trace est éventuellement rejouable, auquel cas, elle devient dynamique. Elle est numérique puisqu'il s'agit d'enregistrements d'actions effectuées sur ordinateur ou d'une version numérisée de vidéo (montrant des humains en interaction ou montrant une capture d'écran lors de cette interaction). ».

Nous adoptons la définition proposée par Lund et Mille en précisant que nous faisons une différence, de par leurs natures différentes, entre (1) une trace d'enregistrements numériques d'actions effectuées sur ordinateur ou à l'aide d'un dispositif technique connecté à un ordinateur et permettant d'effectuer des actions traçables (p. ex. un dispositif haptique ou oculométrique pour le suivi du mouvement oculaire) et (2) une observation humaine collectée manuellement ou un enregistrement audio/vidéo. Le premier type de trace possède une propriété que le deuxième n'a pas, qui consiste en une possibilité de traitement automatique de la trace, en effectuant des calculs et des transformations automatiques sans avoir besoin de l'intervention humaine au début du processus. En effet, une observation humaine, ainsi qu'un enregistrement audio ou vidéo ont besoin d'un travail préalable de codage avant d'être traités par un outil d'analyse permettant d'y effectuer des calculs automatisés. Une raison, peut-être plus profonde de cette distinction, est que cette intervention humaine, aussi objective qu'elle puisse être, constitue déjà en elle-même une analyse ou une interprétation des observés. Nous désignerons donc ces deux types de traces respectivement par traces d'interaction directement traitables par une machine et traces d'interaction non directement traitables par une machine.

En ce qui concerne le caractère temporellement situé des observés de la trace, nous considérons la distinction faite dans (Settouti, 2011) : « … une trace est explicitement composée d'objets arrangés et inscrits par rapport à une représentation du temps de l'activité

tracée. Cet arrangement peut être séquentiel explicite (chaque observé est suivi et/ou précédé par un autre) ou découler d'un estampillage temporel explicite des observés. ».

Dans (Settouti, 2011), la trace d'interaction a fait l'objet d'une formalisation et la notion de trace modélisée est proposée. Un cadre formel, dit système à base de traces modélisées, est défini pour l'exploitation de traces modélisées. Une trace modélisée est définie comme « toute trace issue d'un processus de collecte, composée d'observés temporellement situés, et conforme à un modèle de trace », où modèle est défini comme « le vocabulaire permettant la compréhension de la trace, décrivant abstraitement ses observés et les relations pouvant exister entre eux ». Par ailleurs, Settouti distingue deux types de traces modélisées : (1) traces fermées, qui « n'attendent plus d'ajouts de nouveaux éléments », et (2) traces ouvertes, qui « sont toujours en cours de collecte et ouvertes à l'ajout de nouveaux observés ». Les traces fermées sont utilisées dans un contexte d'analyse à posteriori de l'activité d'apprentissage tracée, tandis que les traces ouvertes sont utilisées pour faire des analyses en temps réel de l'activité durant l'apprentissage afin de permettre à l'apprenant de suivre son activité par des techniques de mirroring, le guider durant son activité ou le conseiller en fonction de ses actions.

Nous nous classons clairement dans le contexte du partage de corpus de traces d'interaction fermées. En effet, les corpus construits sont partagés et analysés d'une manière différée permettant aux chercheurs d'étudier différentes problématiques telles que les processus d'acquisition des connaissances ou la réingénierie des EIAH. Un corpus est considéré fermé à partir du moment où les données correspondantes rentrent dans la base des corpus.

3.3 Contraintes du partage

L'analyse des questions de recherche étudiées a dégagé un ensemble de contraintes à traiter pour étudier la problématique du partage des corpus de traces d'interaction d'apprentissage, des outils d'analyse de traces, et des analyses réalisées.

3.3.1 Hétérogénéité des traces

Lors d'une situation d'apprentissage, les traces enregistrent les interactions entre humains médiatisées par ordinateur et les actions/réactions entre un humain et une machine. Les données enregistrées dans les traces (appelées éléments observés (Settouti et al., 2009))

dépendent des environnements d'apprentissage utilisés. En effet, le traçage de l'utilisation d'un outil de communication tel qu'un chat ou un forum suit un modèle différent de celui d'un tuteur intelligent (ITS). Il est également à noter que le traçage est lié à des choix d'observation liés aux besoins de recherche et d'analyse des chercheurs. Dans (Carron et al., 2005), les auteurs insistent sur l'importance de la phase de pré-expérimentation dans la préparation d'une expérimentation pédagogique. Cette phase de pré-expérimentation permet de définir, en fonction des objectifs d'observation, les facteurs observables pertinents pour des analyses particulières. (Courtin et Talbot, 2007) proposent de procéder à une configuration appropriée des outils d'observation permettant la définition d'observés adaptés aux objectifs d'observation qui varient selon le type d'observateur considéré. (Laflaquière et Prié, 2008) considèrent que les observés sont déterminés pour être pertinents relativement aux objectifs d'exploitation envisagée de la trace. Dans (Choquet et Iksal, 2007), les auteurs considèrent que, dans le cadre de la réingénierie d'un EIAH, le concepteur d'un dispositif d'apprentissage doit définir « ce qu'il est important d'observer » pour identifier ce « qu'il faut collecter ». Par ailleurs, les traces peuvent dans certains cas être spécifiques à un domaine particulier dans le cas de l'utilisation d'un outil d'apprentissage spécialisé (par exemple un tuteur intelligent pour l'enseignement de l'algèbre, comme Aplusix (Nicaud et al., 2002), un tuteur intelligent de sciences physique comme Andes (Gertner et VanLehn, 2000)). L'hétérogénéité des traces est donc due à la diversité des approches et des modèles de traçage choisis par les chercheurs, ce qui produit des traces ayant des formats différents.

3.3.2 Différentes natures des traces

Comme indiqué dans notre définition de trace, nous distinguons deux types de traces : traces d'interaction directement traitables par une machine et traces d'interaction non directement traitables par une machine. Cette différence de nature fait que le traitement de ces traces est différent. En effet, les traces d'interaction non directement traitables par une machine doivent faire l'objet d'une phase de transcription préalable à tout traitement (semi-)automatique possible. Par ailleurs, les deux types de traces sont souvent utilisés conjointement dans les analyses permettant d'avoir des données plus complètes sur le déroulement d'une activité d'apprentissage. Les outils d'analyse des traces peuvent gérer uniquement l'un des deux types, ou travailler sur les deux à la fois, permettant ainsi la synchronisation des deux types de traces comme c'est le cas pour les outils d'analyse Tatiana (Dyke et al., 2009) et ActivityLens (Avouris et al., 2007).

3.3.3 Différents niveaux de granularité des traces

Les traces collectées par des environnements d'apprentissage ont des niveaux de granularité différents. En effet les traces ne sont pas toujours collectées à un niveau facilement compréhensible par un humain (même expert en informatique). Elles peuvent être de niveau très bas et liées aux événements de bas niveau relatifs au dispositif matériel utilisé (par exemple les clics d'une souris, les coordonnées de déplacement de l'œil enregistrées par un oculomètre, les coordonnées de déplacement d'un dispositif haptique). Les traces considérées dans notre travail doivent être à un niveau d'abstraction décrivant des événements faisant sens pour l'activité d'apprentissage instrumentée (par exemple : envoyer un message, répondre à un message, dessiner un objet, répondre à une question, effectuer un acte significatif, etc.). On peut donc imaginer que les traces stockées dans un corpus puissent être des traces transformées provenant d'autres de niveau d'abstraction plus bas.

3.3.4 Nécessité de contextualisation des traces

Il est difficile pour un chercheur de comprendre et d'exploiter des traces provenant d'une expérimentation à laquelle il n'a pas participé surtout si des outils d'apprentissage qu'il ne connait pas ont été employés. Nous soulignons donc l'importance de la contextualisation des traces d'interaction partagées, à l'instar du projet MULCE (Reffay et Betbeder, 2009). L'expérimentation ayant produit le corpus doit être décrite en conservant des ressources en plus des traces, permettant de les contextualiser. Ces ressources contextuelles peuvent contenir des ressources de documentation, des publications ayant traité de l'expérimentation, ainsi que des ressources utilisées ou produites durant l'expérimentation (p. ex. questionnaire présenté à l'apprenant, une production écrite par un apprenant).

3.3.5 Droits d'accès, protection de la vie privée et anonymisation

Le partage de corpus de traces d'interaction pose le problème de la divulgation des données personnelles relatives aux participants à une expérimentation ayant accepté le traçage et/ou l'enregistrement de leur activité. Il est donc important de considérer, pour le partage des corpus, les aspects relatifs aux droits d'accès aux données partagées et à la protection de la vie privée par l'anonymisation des données partagées. Des travaux existants se sont intéressés aux techniques d'anonymisation des données personnelles permettant de prévenir la divulgation de l'identité des personnes tout en gardant les données dans une forme intéressante pour

l'analyse (Reffay et al., 2012) (Reffay et Teutsch, 2007) (Baude et Eshkol, 2007). En tout cas, dans le cadre de ce travail qui constitue une maquette de type « proof of concept », l'étude de cet aspect ne fait pas partie des objectifs de ce travail de thèse. En effet, du point de vue théorique, l'aspect anonymisation n'a pas d'impact sur la modélisation que nous proposons dans ce travail. Quant à la pratique, nous considérons que le chercheur dépose un corpus déjà anonymisé et que ce dernier a été collecté avec le consentement des différents participants à l'expérimentation d'apprentissage qui étaient informés que les données collectées allaient être anonymisées, partagées et analysées strictement dans un objectif de recherche.

3.3.6 Absence d'une représentation standard des traces d'interaction

Nous devons constater l'absence de modèle et format standards universellement reconnus par la communauté des chercheurs du domaine pour la représentation des traces d'interaction. L'existence d'un tel standard aurait résolu le problème de partage d'une formalisation commune des données entre les environnements d'apprentissage et les environnements d'analyse. L'absence d'un tel standard s'explique par la grande diversité dans la sémantique et la granularité des traces collectées (due au fait que les traces sont souvent très liées au domaine étudié et que les données collectées sont parfois choisies en fonction des besoins d'analyse des chercheurs). Dans (Settouti, 2011), une méta-modélisation de la trace est présentée par la notion de trace modélisée. Ce méta-modèle générique propose un vocabulaire permettant de modéliser la trace mais ne définit pas les concepts pouvant exister dans la trace d'interaction d'apprentissage et ne permet pas d'envisager une application pratique. D'autres travaux ((Martínez et al., 2005), (Tutor Message Format, 2013), (Wolpers et al., 2007), (Reffay et al., 2008), (Giguet et al., 2009)), par souci d'interopérabilité, ont proposé des représentations pour permettre le partage. Les solutions proposées restent néanmoins limitées puisqu'il est très difficile de prévoir une représentation répondant aux différents besoins de collecte. Comme souligné dans (Martínez et al., 2005), la proposition d'un format partagé pour la représentation des traces d'interaction suppose un compromis difficile entre (1) un format très générique permettant de représenter une multitude de données mais qui risque de faire perdre la sémantique de certaines données (utiles dans l'automatisation de certains traitements par les outils d'analyse) et (2) un format plus spécifique qui permet d'implémenter des traitements automatiques mais qui restreint la variété des données à représenter.

3.3.7 Couplage fort entre les outils d'analyse et les environnements d'apprentissage

Il est fréquent qu'une équipe de recherche travaillant sur un dispositif d'apprentissage particulier développe un outil d'analyse répondant à ses besoins d'analyse. Le développement d'outils d'analyse est donc souvent fortement lié à l'environnement d'apprentissage (Martínez et al., 2005), et l'outil d'analyse est conçu pour accepter le format de trace généré par l'outil d'apprentissage. Cependant, certains travaux offrent des outils d'analyse plus génériques (par exemple les outils offerts par la plateforme CALICO pour la visualisation des messages échangés dans des forums (CALICO, 2010)). La définition d'un format d'entrée pour l'outil d'analyse reste néanmoins nécessaire.

3.3.8 Nécessité de capitaliser sur les analyses réalisées sur les corpus de traces partagés

Le partage de corpus de traces d'interaction d'apprentissage prend tout son sens lorsque les analyses réalisées sur ces corpus sont partagées et liées aux corpus concernés afin de permettre de consulter, reproduire et enrichir des analyses antérieures. Il faut donc partager le maximum d'informations permettant la reproduction (p. ex. modèle d'interprétation utilisé dans une classification des événements de la trace, les données sur lesquelles l'analyse a été réalisée, etc.). Les outils d'analyse des traces, n'étant généralement pas conçus pour être partagés ou pour produire des ressources partageables, ne produisent pas tous une sortie permettant d'enregistrer le résultat d'une analyse. La sortie d'un outil d'analyse, quand elle est produite, doit pouvoir être liée au corpus concerné. Un modèle de description des analyses doit donc également être proposé comme support au partage entre chercheurs.

3.4 Exemples de corpus

Nous donnons ici trois exemples de corpus de traces collectées par différentes équipes participant au projet « Personnalisation des EIAH » (cluster ISLE, Rhône-Alpes) dans le cadre de leurs travaux de recherche.

- Corpus « EMSE-LEAD » : les traces de ce corpus sont générées par l'environnement DREW (Corbel et al., 2003) et correspondent à l'utilisation d'un chat et d'un tableau blanc lors d'une séance d'encadrement de projet de

programmation. L'analyse de ce corpus a permis d'étudier le rôle des outils offerts par l'environnement DREW dans l'amélioration du dialogue entre apprenants. La structure des traces de ce corpus est définie par une DTD. Les traces sont donc représentées en XML et respectent la DTD définie.

- Corpus forum contextuel (Confor) : les traces de ce corpus (May, 2009) correspondent à l'utilisation au cours d'un module de Français Langue Étrangère d'un forum contextuel (George, 2003) permettant d'organiser les fils de discussion en fonction des activités d'un scénario pédagogique. Les traces collectées ont fait l'objet d'analyses en temps réel en utilisant l'outil TraVis (May, 2009) avec comme objectif d'améliorer l'expérience collaborative des participants à la discussion via des indicateurs visuels calculés en temps réel. Les traces composant ce corpus ont été stockées dans une base de données relationnelle mais un schéma XSD a été défini pour exporter ces données au format XML.

- Corpus Teleos (Luengo et al., 2006) : les traces de ce corpus correspondent à l'utilisation de Teleos, un EIAH utilisé par les étudiants en chirurgie orthopédique connecté à un dispositif haptique simulant le déplacement d'une broche dans le bassin, et dont les coordonnées de déplacement sont collectées. Les traces brutes collectées subissent une première analyse permettant d'identifier la zone du bassin touchée par la broche et ainsi de transformer les traces dans un niveau d'abstraction plus élevé. Ces traces transformées sont également traitées par un module de diagnostic permettant d'identifier les connaissances mises en jeu. Les traces composant ce corpus sont représentées dans un format textuel sous forme de fichier CSV (valeurs séparées par des virgules).

Afin d'évaluer notre approche, nous avons procédé à la construction de deux corpus : (1) le corpus « EMSE-LEAD » mentionné ci-dessus et qui correspond à un corpus existant que nous reconstruisons, (2) et le corpus « COO-POO » qui correspond à une nouvelle expérimentation, réalisée dans le cadre de ce travail de thèse, consistant à évaluer l'utilité de l'utilisation d'un forum de discussion dans l'amélioration de la collaboration entre les participants au module de conception et programmation orientée objet, à l'IUT de Chambéry. Ces deux corpus seront présentés dans le chapitre 9. La construction d'autres corpus, notamment ceux collectés par les équipes du projet EIAH fera l'objet d'un travail futur.

3.5 Quelle approche flexible et générique pour le partage?

Le partage de corpus de traces d'interaction entre communautés de chercheurs utilisant des EIAH n'a pas fait l'objet de beaucoup de travaux de recherche. La trace, ayant très rarement fait l'objet d'étude, est souvent utilisée comme moyen pour l'étude d'autres problématiques. La solution intuitive pour partager des corpus de traces est de respecter une représentation standard. En effet, le respect d'un standard adopté par une communauté peut permettre à un chercheur, ne connaissant pas le corpus, de comprendre son contenu. Le partage d'outils d'analyse sera simple à réaliser s'ils acceptent le même format de données en entrée. Ce besoin de partager des outils d'analyse conforte la pertinence de l'idée d'une représentation standard. Si un tel standard existait, les outils d'analyse seraient adaptés pour recevoir en entrée des données respectant ce standard. Cela permettrait d'accroître l'utilisabilité des outils d'analyse en facilitant l'interopérabilité entre les environnements d'apprentissage et les outils d'analyse. Cependant, un tel standard consensuel n'existe pas pour les raisons que nous avons présentées précédemment (traçage corrélé avec les objectifs d'analyse, différences fondamentales entre les objectifs des environnements d'apprentissage qui sont parfois étroitement liés à un domaine particulier). Cette absence de standard de représentation des traces a motivé des chercheurs à proposer des formalismes pour la représentation des traces utilisés dans la mise à disposition d'outils d'analyse ((Reffay et al., 2008), (Koedinger et al., 2008), (Martínez et al., 2005), (Giguet et al., 2009)). Les formalismes proposés doivent donc nécessairement être respectés pour pouvoir utiliser les fonctionnalités présentées pour le partage et l'analyse. Les problèmes de ces approches résident dans le fait que (1) les formalismes proposés ne peuvent couvrir toutes les données tracées ce qui est dû à l'hétérogénéité des traces pouvant avoir des modèles très variés en fonction des domaines d'application étudiés et des besoins d'observation, et (2) que la conversion des données vers le formalisme proposé peut être un obstacle à l'adhésion des chercheurs qui ne sont pas forcément prêts à investir du temps pour le partage de leurs données en vue de les analyser avec les outils d'analyse offerts ou de les mettre à la disposition d'autres chercheurs.

Il est pertinent, afin de situer notre approche par rapport aux réalisations existantes, de rappeler les objectifs de notre travail de recherche. Notre finalité est de pouvoir rassembler, sous une forme opérationnelle, un ensemble de corpus, et d'être capable de les utiliser pour

réaliser un ensemble d'analyses. Les corpus préexistent à la conception du système et sont hétérogènes : ils ont été créés par des équipes de recherches différentes, avec des outils différents, dans des finalités différentes. Les corpus ont fait ou vont faire l'objet d'analyses par les chercheurs qui les ont construits. Là encore, l'hétérogénéité des outils d'analyse et des analyses est forte. Ce qui unifie ces corpus est le fait qu'à côté d'autres données, ils rassemblent des traces d'interactions (que nous définissons comme des séquences sensiblement homogènes d'événements temporellement situés).

Nous ne sommes donc pas face à la problématique "habituelle" d'une équipe de recherche, qui consiste à concevoir et à construire ex nihilo un futur système, répondant à un cahier des charges précis et cohérent (mais éventuellement négociable avec les chercheurs concernés), permettant d'aboutir à un modèle clair et simple, lequel peut ensuite être utilisé pour démontrer la conformité de la réalisation aux spécifications originelles. Un aspect de notre recherche consiste à prendre en compte l'ensemble des spécifications des corpus à rassembler, et l'ensemble des outils d'analyses utilisés par les chercheurs participant au projet. Cette étude doit fournir les caractéristiques d'une infrastructure unique, pouvant gérer correctement l'ensemble des données impliquées dans la démarche. Ce travail lui-même, bien qu'indispensable, n'est qu'un élément de notre démarche, puisque la finalité de notre recherche est de permettre l'interopérabilité des outils d'analyses sur les différents corpus représentés dans la base.

Ceci nous conduit à expliciter l’un des objectifs de notre travail de recherche : le problème n'est plus vraiment la définition d'une représentation idéale des corpus (les corpus préexistent) ni le choix ou la définition d'outils d'analyses (ils ont déjà été choisis par les chercheurs), mais dans l'interface qui va permettre cette interopérabilité des outils sur les corpus.

Dans notre cas, du fait de l'hétérogénéité des sorties (ce que l'on sait extraire d'un corpus donné) et des entrées (ce qu'attendent les outils d'analyse), il n'y a pas de bonnes raisons de faire reposer la conception de l'interface sur un format spécifique de corpus, qui décrirait à la fois la sémantique et la représentation des données. Au contraire, il semble plus pertinent d'axer l'étude sur la notion de services attendus par les outils d'analyse. Cette hétérogénéité des corpus représentés, et la volonté de mutualiser les outils d'analyse suggèrent donc l'utilisation d'un "intermédiaire", capable d'extraire les données pertinentes d'un corpus, et de les fournir dans la représentation attendue par ces outils.

En simplifiant la situation, nous considérons que les entrées de nos outils d'analyse consistent en séquences homogènes d'ensembles de données temporellement situées, chaque donnée pouvant être définie par un couple sémantique/représentation, noté (S, R). Le challenge est dès lors de pouvoir définir les couples (S_a, R_a) attendus par l'outil d'analyse en fonction des couples (S_c, R_c) disponibles dans le corpus.

La description des sémantiques des données intervenant dans l'interface devenant d'une importance essentielle dans notre approche, nous faisons le choix de décrire ces sémantiques au travers d'une taxonomie de concepts faisant partie d'une ontologie descriptive qui conceptualise le système, qui sera à la fois un outil de référence et de travail pour le chercheur, et un support informatique pour le système.

Puisque nous souhaitons automatiser autant que faire se peut le mécanisme d'extraction de données du corpus et de création d'entrées pour les outils d'analyse, nous utiliserons un composant logiciel qui, en s'appuyant sur l'ontologie, sera capable de réaliser ce travail. Nous avons baptisé « proxy » ce composant, en ce sens qu'il se comporte comme un représentant du chercheur dans son travail de sélection et de transformation de données. Le « proxy » travaille par sélection et assemblage de programmes élémentaires préexistant, dont l'exécution va réaliser les transformations requises pour fournir les entrées aux outils d'analyse.

Le point d'attaque de notre approche est dès lors clair : nous conservons les corpus dans leur format d'origine, nous fournissons une taxonomie des concepts intervenant dans les corpus (ou, en tout cas, une taxonomie de base pouvant être étendue en fonction des besoins des chercheurs), et une infrastructure logicielle spécialisée effectue les opérations de conversion nécessaires aux analyses. Notre contrainte, faible, est que nous nous trouvons dans un univers XML, et qu'un moteur de requêtes XQuery doit être disponible. De fait, la quasi-totalité des outils d'expérimentation et de recueil de traces génèrent effectivement leurs données dans cette représentation ; dans les autres cas, il est presque toujours immédiat de choisir un dtd ou un XSchema, et de construire un petit outil (avec Lex, Yacc et C) pour représenter en XML un fichier de traces qui n'est pas fourni dans ce formalisme. Enfin, il existe de multiples entrepôts de données XML fournissant également un service XQuery.

Donnons quelques exemples de cette approche, que nous baptiserons « Proxyma » (pour « Proxy for Multiple Analyses »).

1. On désire intégrer un nouveau corpus au système. Deux ensembles de données sont à fournir. D'une part un ensemble de données descriptives permettant de documenter le corpus (nous reviendrons ultérieurement sur cet aspect). D'autre part, une description de la sémantique des diverses données contenues dans le corpus. On peut désirer ne décrire qu'une partie de ces données (par exemple, on peut ne pas s'intéresser à la trace des adresses IP des machines ayant été utilisées lors de l'expérimentation). Le réel travail consiste à répertorier les types de données auxquelles on veut donner accès, à associer à ces données une signification connue dans la taxonomie de concepts définie dans l'ontologie des corpus, et enfin à expliquer comment accéder, dans le corpus, à ces données, par exemple, en associant à ces types de données un chemin d'accès (XPath/XQuery) dans le corpus, et en décrivant leur représentation. S'il n'est pas possible d'associer une sémantique préexistante dans le système à une donnée spécifique du corpus, il est possible de compléter a taxonomie pour ajouter ce concept, et permettre la description des données du corpus.

2. Un chercheur veut effectuer une analyse A sur un corpus C au moyen de l'outil O. Le corpus a été intégré à notre base de corpus, et la sémantique de ses données décrite ; de même, l'outil O est connu du système et la sémantique des données attendues également. L'analyse suppose que O reçoive ses entrées dans un certain format F. La création de l'entrée utile à l'analyse A nécessite, pour chaque couple (S_i, R_i), de trouver un chemin de conversion permettant d'obtenir une donnée de sémantique S_i à partir de données de sémantiques S'_1, S'_2, etc. disponibles dans la base de corpus. La correspondance peut être immédiate, ou peut être obtenue par une transformation au moyen d'un script XQuery déjà existant, ou, hélas, devant être écrit pour l'occasion. L'étape précédente nous construit un couple (S_i, R'_i). Si la représentation R'_i n'est pas identique à R_i, un autre script de conversion doit être mis en œuvre.

3. On désire décrire un outil d'analyse. Le travail exact à accomplir dépend énormément des spécifications de l'outil considéré. Dans le cas de Tatiana (Dyke, 2009), par exemple, qui n'a pas été conçu pour analyser directement un type particulier de corpus, il existe un format d'entrée abstrait propre à cet outil (ou, plus précisément, une classe de formats, similaires, mais pouvant différer par la nature des données représentées), qui a la sémantique d'une trace d'interaction. Dans l'utilisation actuelle de Tatiana, des scripts XQuery, permettant d'extraire des données depuis un corpus et de les convertir dans le format attendu par Tatiana, sont construits au coup par coup par le chercheur,

en fonction des besoins. On peut imaginer que l'approche Proxy+Ontologie peut simplifier, voire automatiser la rédaction de ces scripts. D'autres outils d'analyse (CALICO (CALICO, 2010), PSLC Datashop (PSLC Datashop, 2013)) ont été conçus pour un format spécifique de corpus. Ils vont donc s'appliquer immédiatement aux corpus de ce type, mais la description (S, R) des données attendues pour une analyse permettra, comme évoqué au (2) ci-dessus, de réaliser des analyses sur d'autres types de corpus.

Tout en restant dans le cadre relativement restreint de la gestion des corpus de traces d'interactions, l'approche « Proxyma » est donc susceptible de proposer aux chercheurs simplicité, souplesse et ouverture. Ainsi, si une équipe a construit un corpus C_1, et a réalisé une analyse A_1 de ce corpus au moyen d'un outil d'analyse O_1, après intégration de C_1 et O_1 dans le système, la même analyse peut être réalisée sur le corpus intégré aussi simplement que sur le corpus brut, avec un résultat identique. L'apport de Proxyma est que le résultat de l'analyse A_1 va pouvoir à son tour être associé à C_1 et intégré à la base de corpus.

Il faut enfin rappeler que la finalité de notre recherche est d'aboutir à la spécification d'un système offrant les propriétés désirées. Nous nous situons donc moins sur le plan d'une modélisation formelle, abstraite, d'un tel système, que sur le plan du génie logiciel, visant à décrire les composants du système, leurs propriétés, et leur mise en œuvre.

En résumé, nous proposons une nouvelle approche de partage des corpus de traces d'interaction d'apprentissage et des outils d'analyse. L'idée principale sous-tendant notre proposition est d'éviter de passer par encore une nouvelle représentation des traces imposée, laquelle devant être respectée pour permettre ce partage. Cette approche ayant démontré ses limites, nous nous proposons de ne pas imposer de formats spécifiques pour la représentation des ressources pouvant être retrouvées dans un corpus, et proposons l'approche « Proxyma », une nouvelle approche flexible et générique. Cette approche consiste à accueillir toute ressource, jugée par le chercheur comme étant pertinente pour le partage et l'analyse, dans son formalisme original et de définir un modèle basé sur une ontologie et qui remplit trois fonctions principales en définissant :

- un modèle de corpus, définissant les types de ressources partageables dans un corpus, et le modèle de description du corpus et de ses composants,

- un modèle opérationnel définissant les mécanismes permettant l'interrogation des corpus,

- un modèle sémantique définissant un ensemble de concepts généralement retrouvés dans les traces d'interaction permettant l'alignement sémantique entre ces concepts et ceux de l'ontologie implicite (existante dans l'esprit du chercheur) qui définit les concepts retrouvés dans les traces d'un corpus partagé.

Le fait de ne pas imposer un formalisme spécifique pour la représentation des traces, permet non seulement un investissement initial minimal de la part du chercheur, mais aussi de préserver la richesse sémantique des données hétérogènes collectées qui risque d'être altérée suite à une conversion. En nous basant sur cette approche, nous avons proposé une architecture de plateforme « Beatcorp » pour le partage et l'analyse de corpus de traces contextualisées.

Les chapitres suivants traiteront de la présentation des trois modèles de l'approche « Proxyma » et de l'évolution de ces modèles, de l'architecture de la plateforme « Beatcorp », et des exemples d'application de notre approche.

Chapitre 4 : L'approche « Proxyma » : modèle de Corpus

4.1 Introduction

Dans ce chapitre, nous présentons le premier modèle de l'approche « Proxyma ». Il s'agit du modèle de corpus permettant la modélisation d'un corpus. Ce modèle est essentiel pour permettre le partage d'un corpus et l'interrogation de sa description. Il nous permet d'expliciter les types de ressources partagées dans un corpus ainsi que les différentes

métadonnées nécessaires pour la description générale du corpus et des ressources qui le composent ce qui permet de contextualiser les traces. Ce modèle permet également de décrire les analyses réalisées et d'établir le lien analyse/corpus. Ce modèle constitue le premier élément essentiel de l'approche « Proxyma ». En effet, un chercheur souhaitant effectuer une analyse sur un corpus qu'il ne connait pas a besoin d'accéder à la description du corpus soit en consultation soit en exécutant des requêtes prédéfinies. Par ailleurs, un chercheur souhaitant profiter de la possibilité de reproduire une analyse offerte par l'approche « Proxyma » a besoin d'accéder à (1) la description de l'analyse et des objectifs l'ayant motivée, (2) l'interprétation du chercheur l'ayant réalisée, et éventuellement (3) les publications qui s'y rapportent. De plus, l'approche « Proxyma » étant évolutive et participative, un chercheur peut devoir consulter la description du corpus pour créer un nouveau script (modèle opérationnel) lui permettant d'expliciter le lien entre la taxonomie de concepts (modèle sémantique) définie dans l'ontologie et les incarnations de ces concepts que l'on trouve dans le corpus.

Nous commençons ce chapitre par l'introduction du concept d'expérimentation d'apprentissage qui représente un outil de validation scientifique des modèles et outils en EIAH. En effet, dans cette démarche, les chercheurs conçoivent une expérimentation d'apprentissage. La réalisation d'une telle expérimentation leur permet alors de construire des corpus de données (traces et autres) correspondant à l'utilisation des environnements d'apprentissage. Enfin, les chercheurs analysent ces corpus pour évaluer leurs modèles. Après la présentation de ce qu'est pour nous une expérimentation d'apprentissage, la notion de corpus est définie et deux types de corpus sont proposés. Un corpus est composé de ressources physiques et possède une description de ses différents composants. Les différents types de ressources partageables dans un corpus ainsi que le modèle de description d'un corpus sont donc présentés.

4.2 Qu'est-ce qu'une expérimentation d'apprentissage ?

Nous présentons dans ce paragraphe la notion d'expérimentation d'apprentissage. En effet, dans l'objectif de valider leurs modèles et outils, les chercheurs réalisent des expérimentations d'apprentissage leur permettant de construire des corpus de données analysables. L'expérimentation est un outil de validation scientifique utilisé par les chercheurs

afin de valider leurs hypothèses de recherche. Elle représente un moyen d'observation pour le chercheur souhaitant étudier une problématique. Dans le domaine des EIAH, la réalisation d'expérimentations est une pratique très fréquente. Une expérimentation d'apprentissage, suivant les objectifs du chercheur, peut permettre l'observation dans une optique (1) d'ingénierie/réingénierie d'un EIAH (Choquet, 2007) (Mostow et Aist, 2001), (2) d'étude du rôle des outils offerts par l'environnement d'apprentissage dans la facilitation de l'apprentissage (May, 2009) (Di Eugenio et al. 2005), (3) d'analyse des profils d'apprenants et des processus d'acquisition des connaissances (Luengo et al., 2006) (Koedinger et al., 2008) (Tao et al., 2007), (4) de validation du déroulement d'un scénario pédagogique effectif par rapport à un scénario prédictif (Ferraris et Lejeune, 2009) (Wolpers et Grohmann, 2005). Une expérimentation peut soit correspondre à une situation écologique, auquel cas, une situation d'apprentissage réelle a lieu, soit correspondre à une expérience de laboratorie effectuée par un chercheur avec son équipe de recherche, pour des validations préliminaires. Une situation écologique est généralement beaucoup plus complexe et consomme beaucoup plus de temps. Une expérimentation, dans notre contexte, consiste à concevoir une situation d'apprentissage utilisant un EIAH, éventuellement formalisée à l'aide d'un langage de modélisation pédagogique (par exemple IMS-Learning Design (IMS-LD, 2003), Learning Design Language (Martel et al., 2006)), et dont l'observation génère des corpus de traces d'interaction. Le domaine des EIAH étant pluridisciplinaire, les chercheurs impliqués dans une expérimentation peuvent avoir des problématiques de recherche différentes. Par exemple, nous pouvons imaginer que des chercheurs en linguistique soient intéressés par l'étude des traces d'interaction collectées d'un point de vue linguistique, alors que des chercheurs en technologies de communication soient intéressés par l'étude de l'utilité des fonctionnalités de communication offertes aux utilisateurs. Par ailleurs, dans le cas d'une expérimentation relative à une situation pédagogique écologique, les objectifs pédagogiques d'une expérimentation peuvent être dissociés des problématiques de recherche. Par exemple, une expérimentation conçue dans le cadre de l'enseignement d'un module de langue permet à un chercheur en ingénierie des EIAH d'évaluer l'environnement d'apprentissage et les outils qu'il offre afin d'identifier les points à améliorer, et à un autre chercheur d'évaluer les fonctionnalités offertes par un outil d'analyse des interactions. Une expérimentation peut donc servir dans l'étude de problématiques de recherche différentes de celles pour lesquelles elle a été conçue.

4.3 Modèle de corpus

Cette section décrit le modèle de corpus que nous proposons pour le partage de corpus de traces d'interactions contextualisées ainsi que les analyses réalisées sur ces corpus. Ce modèle permet la spécification de la description des corpus et de leurs contenus. Une telle description est nécessaire pour la gestion des corpus, leur interrogation, et leur analyse. La description d'un corpus, intégrant une partie de description des analyses réalisées sur le corpus, permet à d'autres chercheurs d'accéder aux analyses antérieures, de les répliquer, vérifier, valider, et éventuellement les enrichir. Les chercheurs peuvent reprendre des analyses déjà réalisées pour étudier de nouvelles questions de recherche. La modélisation et le partage de corpus peut également s'avérer très bénéfique pour les jeunes chercheurs, devant valider leurs modèles et ne disposant pas d'assez de temps ou de moyens pour monter une expérimentation et structurer les données collectées. Il est également à noter que le partage de corpus entre chercheurs peut aboutir à des corpus de référence servant au benchmarking et permettant aux chercheurs de comparer leurs modèles, leurs outils d'analyse et la pertinence de leurs résultats.

4.3.1 Qu'est-ce qu'un corpus ?

Comme nous l'avons souligné dans le paragraphe 3.3.4, il est nécessaire de contextualiser les traces d'interaction partagées dans un corpus pour qu'elles soient intelligibles pour un chercheur ne connaissant pas la situation d'expérimentation (Reffay et al., 2008). Cette contextualisation est réalisée via une description du corpus permettant de décrire son contenu et l'expérimentation l'ayant produit. Par ailleurs, des ressources, autres que les traces, peuvent être partagées dans un corpus pour permettre leur contextualisation. Un corpus est donc constitué d'un ensemble de ressources[9] de différents types et d'une description du corpus récapitulative de ses caractéristiques et de son contenu permettant une interrogation uniforme de la base de corpus partagée. Le modèle de description d'un corpus, ainsi que les types de ressources partageables seront présentés dans la suite.

[9] Nous désignons par ressources tous fichiers informatiques pertinents pour la situation, fichiers qui peuvent avoir différents formats selon les données qu'ils contiennent et l'information qu'ils véhiculent.

4.3.2 Deux types de corpus

Nous distinguons deux types de corpus (cf. Figure 30). Le premier type « *corpus initial* » contient les ressources relatives à une expérimentation et collectées auprès du chercheur souhaitant rendre disponible un corpus. Il contient toutes les ressources collectées et produites avant le partage du corpus. Le deuxième type « *corpus d'analyse* » désigne un corpus lié à une question de recherche particulière et contenant des travaux d'analyse réalisés sur un ou plusieurs corpus à l'aide d'un ou plusieurs outils d'analyse. Cette différenciation nous permet de séparer (1) les ressources initiales (traces + ressources contextuelles) collectées et fournies par le chercheur, ainsi que les ressources relatives aux travaux d'analyse réalisés sur le corpus en dehors de la plate-forme de partage, et (2) les travaux d'analyses réalisées au sein de la plate-forme de partage à l'aide des outils d'analyse partagés.

4.3.2.1 Corpus initial

Un corpus initial (cf. Figure 30) est le résultat de la conception et de l'observation d'une expérimentation d'apprentissage instrumenté par un EIAH. Un corpus initial est construit en :

- Fournissant l'ensemble des ressources qui le composent :
  - ressources correspondant aux traces d'interaction collectées ;
  - ressources pédagogiques contextuelles fournies dans l'environnement d'apprentissage ;
  - ressources produites par les participants durant l'apprentissage ;
  - ressources de documentation du corpus et de ses contenus (p. ex. description du corpus, les droits d'accès aux ressources) ;
  - ressources relatives aux analyses réalisées sur le corpus : ressources utilisées et produites durant les travaux d'analyse réalisés sur les traces, et ressources d'interprétation des résultats obtenus dans les ressources produites ;
  - et ressources de types publications liées au corpus et aux travaux d'analyse réalisés sur le corpus.
- Fournissant les informations nécessaires à la description du corpus, les ressources qui le composent, et les travaux d'analyse déjà réalisés sur le corpus en dehors de la plate-forme de partage.

Un corpus initial peut référencer des travaux d'analyse, réalisés sur le corpus en dehors de la plate-forme, avant ou après la mise en place du corpus. En effet, un travail d'analyse

ayant utilisé un ou plusieurs outils d'analyse non partagés dans la plate-forme peut être décrit et intégré au corpus initial. Par ailleurs, si un chercheur utilise un outil d'analyse partagé dans la plate-forme, mais n'utilise pas les moyens offerts par la plate-forme pour l'extraction et la conversion des données (i.e. le couplage entre les données du corpus et l'outil d'analyse est réalisé indépendamment de la plate-forme), l'analyse en question peut être intégrée au corpus initial.

4.3.2.2 Corpus d'analyse

Un corpus d'analyse (cf. Figure 30) est construit dans la plate-forme de partage dans l'objectif d'étudier une question de recherche particulière intéressante pour un chercheur ou une équipe de recherche. Un corpus d'analyse fait référence à des ressources, qu'il réutilise dans les travaux d'analyse qui le composent, et qui proviennent d'un ou plusieurs corpus initiaux et/ou d'analyse déjà partagés (et qui peuvent être des ressources pédagogiques, des ressources de type traces, et des ressources importées et/ou produites par les travaux d'analyse intégrés aux corpus référencés). Par ailleurs, un corpus d'analyse contient des ressources d'analyse. Ces ressources peuvent être : (1) des ressources complémentaires importées pour la réalisation d'un travail d'analyse, (2) des ressources produites par les outils d'analyse résultant du travail d'analyse, et (3) des ressources d'interprétation expliquant les ressources produites. Il peut aussi contenir des ressources de type publication et des ressources de documentation.

Comme pour un corpus initial, un corpus d'analyse est représenté par une description générale du corpus, des ressources qui le composent et des travaux d'analyse réalisés sur ce corpus. Il n'y a pas de contraintes concernant le nombre des travaux d'analyse pouvant être intégrés à un corpus d'analyse tant qu'ils sont liés à la question de recherche étudiée par ce corpus d'analyse. La construction d'un nouveau corpus d'analyse étant, comme déjà signalé, liée à l'étude d'une question de recherche particulière. Un travail d'analyse peut être réalisé sur un ou plusieurs corpus partagés. Un chercheur peut être intéressé par un travail d'analyse antérieur ou par une ressource contenue dans un corpus d'analyse précédemment construit. Un corpus d'analyse peut donc référencer selon les besoins un ou plusieurs corpus initiaux et/ou d'analyse.

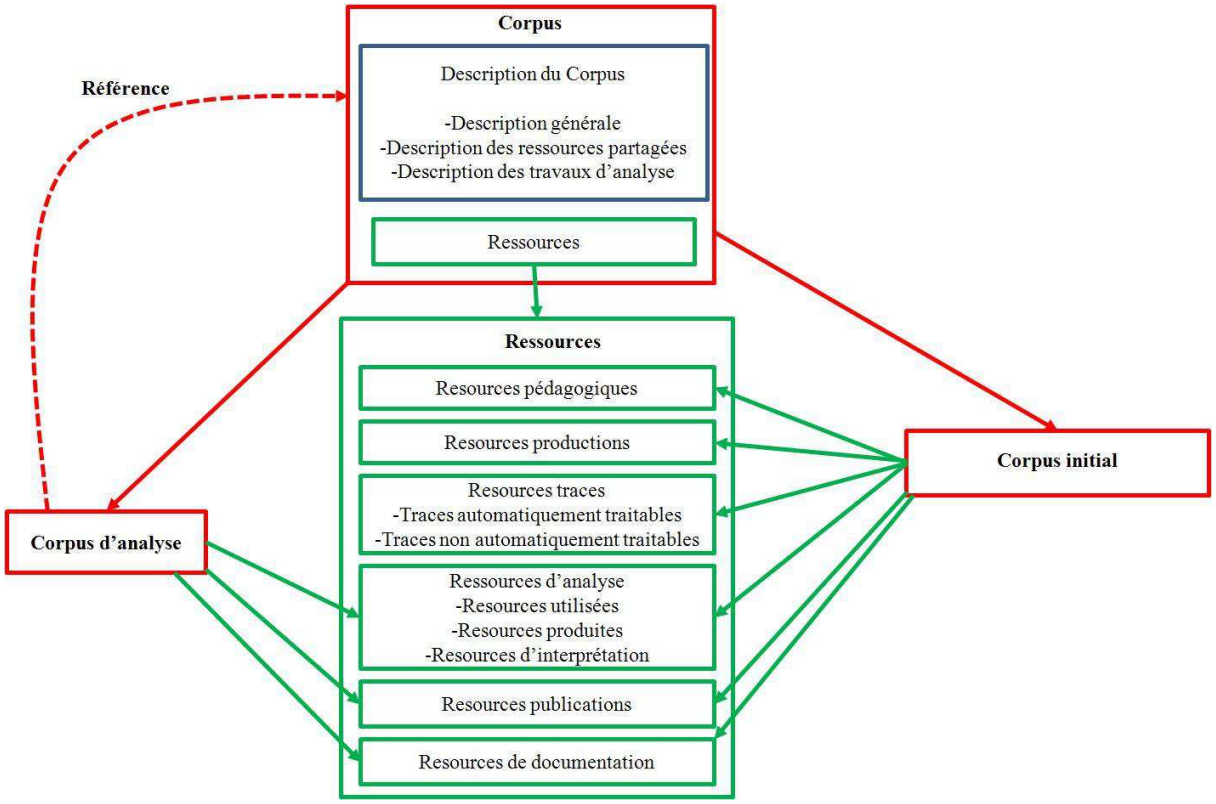


Figure 30 Structure des corpus partagés : corpus initial et corpus d'analyse

4.3.3 Ressources partagées dans un corpus

Pour assurer la contextualisation des traces d'interaction partagées dans un corpus, nous considérons, à part les traces, cinq types de ressources partageables dans un corpus (cf. Figure 30). Suivant qu'il s'agit d'un corpus initial ou d'un corpus d'analyse, les types des ressources partagées varient. Nous présentons les différents types de ressources et justifions leur utilité.

4.3.3.1 Ressources de traces

Les corpus relatifs à des expérimentations d'apprentissage construits et partagés par des chercheurs pour des chercheurs sont centrés sur les traces. Comme déjà présenté dans la définition des traces considérée dans notre travail, les traces correspondent à des ressources directement traitables par la machine (p. ex. des fichiers de logs) et des traces non directement traitables (p. ex. des enregistrements audio/vidéo qui peuvent subir des traitements préalables permettant de les structurer en utilisant une représentation informatique différente, typiquement une transcription et/ou annotation). Les ressources traces partagées dans les

corpus, et enrichies par les ressources contextuelles, seront interrogées en se basant sur les deux autres modèles de l'approche « Proxyma ».

4.3.3.2 Ressources pédagogiques

Les ressources pédagogiques concernent le contexte d'apprentissage. Une ressource pédagogique est offerte à un participant durant l'apprentissage. Elle peut contenir l'explication de l'activité à accomplir, un support de cours mis à la disposition de l'apprenant, l'énoncé d'un problème à résoudre, etc. La Figure 31 illustre quelques types de ressources pédagogiques que nous avons recensées, parmi lesquels certains sont définis par le standard LOM (Learning Object Metadata) (LOM, 2002) comme types d'objets pédagogiques. Ces types recensés ne sont pas exhaustifs et peuvent être enrichis au besoin.

Le partage des ressources pédagogiques permet à un chercheur qui explore un corpus de mieux comprendre les traces. En effet, le seul partage des traces d'interaction peut ne pas suffire au chercheur pour comprendre ce qui s'est passé. Par exemple, s'il ne connaît pas l'énoncé du problème sur lequel un apprenant a travaillé, ou les supports de cours qui lui ont été fournis, un chercheur ne peut pas avoir une idée claire sur l'activité d'apprentissage proposée à l'apprenant.

Types des ressources pédagogiques
Scénario pédagogique
Tutoriel
Livre électronique
Cours
Article
Glossaire
Résolution d'un problème
Types définis par le standard LOM
Exercice
Questionnaire
Diagramme
Figure
Graphe
Diaporama
Tableau
Examen
Expérience
Enoncé de problème

Figure 31 Quelques types de ressources pédagogiques contextuelles

4.3.3.3 Ressources de production

Une ressource de production est produite par un participant au cours de son activité d'apprentissage. Elle peut être la solution à un problème, une production écrite, etc. La Figure 32 illustre quelques types possibles de ressources de production. Le partage de ces ressources permet au chercheur de disposer, à côté des traces, des artefacts produits par l'apprenant durant l'apprentissage. En particulier, ce partage permet au chercheur de disposer du produit de l'activité ainsi que les étapes ayant permis d'y parvenir (contenues dans les traces d'interaction). Comme expliqué dans (Settouti, 2011), une ressource de production peut, dans le cadre d'une définition générale où « *une trace est la marque laissée par une activité* », être considérée comme une trace puisqu'elle représente une marque de l'activité d'apprentissage d'un participant. Comme indiqué dans le paragraphe 3.2, nous ne considérons comme traces que celles qui contiennent des marques temporelles permettant de suivre la succession des événements pour l'accomplissement d'une activité d'apprentissage.

Types des ressources productions
Diagramme
Essai
Figure
Graphe
Réponses questionnaire
Résolution d'un problème
Diaporama
Résumé
Tableau

Figure 32 Quelques types des ressources productions

4.3.3.4 Ressources d'analyse

Les ressources d'analyse concernent l'activité d'analyse des traces d'interaction d'apprentissage collectées, réalisée a posteriori de l'activité d'apprentissage par des chercheurs pour répondre à des questions de recherche spécifiques. L'analyse peut être réalisée en dehors de la plate-forme de partage ou dans le cadre de la plate-forme, à l'aide des outils partagés et utilisant les mécanismes d'interrogation offerts. Nous avons distingué trois types de ressources liées à l'analyse : (1) ressources utilisées durant l'analyse, (2) ressources produites par l'analyse, et (3) ressource d'interprétation des résultats d'analyse. Les traces sont souvent insuffisantes pour un travail d'analyse. En effet, un chercheur peut avoir besoin de données complémentaires utilisées pendant l'analyse. Par exemple, dans le cadre d'un

diagnostic cognitif des connaissances d'un apprenant à partir des traces d'interaction, un modèle représentant les connaissances du domaine est nécessaire pour l'analyse des traces (Luengo et al., 2006). Un autre exemple serait un modèle de catégorisation des événements d'interaction (p. ex. le cadre analytique Rainbow (Baker et al., 2007) qui offre un schéma de codage des débats entre apprenants dans un environnement d'apprentissage collaboratif assisté par ordinateur).

L'analyse des traces d'interaction à l'aide d'un outil d'analyse peut générer des ressources produites par le processus d'analyse. Quand cela est possible, il est important de partager ces ressources. Le partage des ressources utilisées et des ressources produites permettent la contextualisation de l'analyse et sa reproduction. Mais elle peut également permettre la comparaison de deux analyses concurrentes (p. ex. utiliser la méthode des juges (inter-coder reliability) pour appliquer un même schéma de codage sur les mêmes traces par deux chercheurs différents et comparer les résultats (De Wever et al., 2006)). Nous avons identifié quelques types des ressources utilisées (cf. Figure 33) et produites (cf. Figure 34). Ces types recensés ne sont pas exhaustifs et peuvent être enrichis si besoin est. Enfin, lors de son activité d'analyse, un chercheur peut rédiger des documents de travail contenant des interprétations des résultats du processus d'analyse. Le partage de ces interprétations est intéressant pour un chercheur qui exploite un corpus partagé, il peut vérifier s'il est d'accord ou non avec ces interprétations et, éventuellement, se baser sur ces interprétations pour en proposer de nouvelles.

Types de ressources d'analyse utilisées
Questionnaire
Interview
Profil de l'apprenant
Modèle de connaissances
Modèle de catégorisation

Figure 33 Quelques Types des ressources d'analyse utilisées

Types de ressources d'analyse produites
Scénario d'analyse
Transcription
Annotation
Catégorisation
Graphe
Calculs et statistiques
Profil de l'apprenant

Figure 34 Quelques types des ressources produites par les analyses

4.3.3.5 Ressources de type publication

La valorisation d'un travail de recherche scientifique est liée, entre autres, aux publications relatives au travail réalisé. Les chercheurs, pour étudier une problématique, montent une expérimentation qui leur permet de construire un corpus. Ce corpus leur sert de base pour leurs analyses. Une pratique courante chez les chercheurs consiste en la publication d'articles scientifiques décrivant l'expérimentation, ses objectifs, les analyses et les résultats obtenus. Le partage des publications relatives aux corpus et aux analyses réalisées sur ces corpus permet une bonne contextualisation des traces et des analyses réalisées. Les publications peuvent être de différents types (cf. Figure 35).

Types de publications
Conférence
Revue
Chapitre d'ouvrage
Ouvrage
Atelier
Séminaire
Poster
Rapport technique
Thèse

Figure 35 Quelques types des publications

4.3.3.6 Ressources de documentation

Un chercheur travaillant sur une expérimentation et l'analyse du corpus des traces collectées peut produire de la documentation décrivant l'expérimentation, les objectifs d'apprentissage, les objectifs de recherche, la description d'un travail d'analyse, la licence d'utilisation associée au corpus ou les ressources qui le composent, etc. Les ressources de documentation permettent de renseigner sur le contenu d'un corpus partagé.

4.3.4 Exemple

Nous donnons un exemple d'un corpus relatif à une expérimentation d'apprentissage écologique réalisée à l'Université de Savoie (Talbot et Courtin, 2008). Cet exemple nous permet d'illustrer les différents types de ressources pouvant être partagées dans un corpus. L'expérimentation met en place une situation d'apprentissage collaborative où des dyades d'étudiants, dans le cadre du cours de langue anglaise, définissent des termes anglais choisis

dans un texte fourni par un enseignant. Chaque apprenant de chaque dyade travaille sur une machine et doit identifier les mots à définir. Les apprenants de la dyade peuvent communiquer entre eux en utilisant un outil de chat. La production de définitions se fait quant à elle dans un éditeur de texte partagé. Quand une dyade finit la définition d'un terme, cette définition est soumise pour correction et/ou validation par l'enseignant. Ce dernier consulte alors la définition, l'annote pour rectification si elle est incorrecte, ou la valide si elle est correcte. Au besoin les étudiants peuvent demander de l'aide à l'enseignant via l'outil de chat. Les outils utilisés sont instrumentés pour permettre la collecte automatique de traces d'utilisation de l'éditeur de texte partagé et des traces de communication dans l'outil de chat. Des observateurs humains étaient présents durant la séance d'apprentissage pour observer l'activité des dyades. Ils prenaient des notes d'observation manuellement (en utilisant un papier et un stylo). Les données collectées durant cette expérimentation ont été analysées pour évaluer le rôle de la station d'observation CARTE dans l'amélioration du suivi de la séance d'apprentissage par l'enseignant. Les données relatives à l'expérimentation n'étant pas conservées par les chercheurs concernés, elles ont malheureusement été supprimées suite à une mise à jour du serveur hébergeant ces données. La construction d'un corpus relatif à cette expérimentation suivant le modèle de corpus que nous proposons étant donc impossible, nous utilisons cette expérimentation écologique pour donner un exemple.

Un corpus initial correspondant à cette expérimentation contiendrait :

- Ressources pédagogiques : l'énoncé expliquant aux dyades l'activité d'apprentissage et ses différentes phases, ainsi que le texte contenant les termes à définir.
- Ressources productions : le corpus des définitions produites par les dyades.
- Ressources traces :
  - Traces directement traitables par une machine : traces de communication dans le chat relatives aux discussions entre les étudiants des différentes dyades durant leur travail de définition ainsi que les communications avec l'enseignant, traces de l'éditeur de texte partagé relatives à l'activité de définition des termes, traces de l'activité de l'enseignant qui supervise la situation d'apprentissage, et traces des annotations effectuées par l'enseignant sur les définitions soumises par les dyades. Il y a enfin les traces produites par

l'analyseur qui correspondent aux fonctionnalités augmentées pour prendre en compte la régulation de l'activité.

- Traces non directement traitables par une machine : les notes d'observations prises manuellement par les observateurs humains. Ces ressources, pour être stockées sur un support informatique, doivent être saisies, et éventuellement être transcrites dans un format structuré permettant leur croisement et synchronisation avec les autres traces.

– Ressources de documentation : documents relatifs à la description de l'expérimentation.

En supposant qu'un corpus initial contenant ces différentes ressources soit partagé dans la plateforme, des analyses peuvent être réalisées sur les ressources de ce corpus. L'outil d'analyse Tatiana (Dyke, 2009) peut être utilisé pour analyser tout ou une partie des traces collectées. Par exemple, il peut être utilisé pour transcrire les traces d'observation, et produire une ressource contenant le résultat de la transcription. Après la conversion des traces dans le format adéquat (display format), Tatiana permet aussi de faire le lien entre le contenu des interactions dans le chat et des définitions écrites dans l'éditeur de texte collaboratif. Un corpus d'analyse, construit pour étudier une question de recherche particulière peut être construit et contenir les ressources produites par l'analyse. Une ressource d'analyse peut être un graphe illustrant les liens entre l'activité de définition dans l'éditeur de texte partagé et les interactions de communication entre les apprenants dans le chat.

4.3.5 Modèle de description d'un corpus

Un corpus partagé est, comme déjà expliqué, formé (1) d'un ensemble de ressources physiques de différents types et (2) d'une description composée d'un ensemble de métadonnées générales décrivant le corpus, d'une description des ressources composant le corpus, et d'une description des travaux d'analyse contenus dans le corpus. La description d'un corpus initial peut contenir la description des travaux d'analyse réalisés sur le corpus en dehors de la plate-forme de partage. Par ailleurs, les travaux d'analyse réalisés au sein de la plate-forme à l'aide des outils d'analyse mis à disposition sont décrits dans des corpus d'analyse créés chacun pour répondre à une question de recherche particulière.

4.3.5.1 Métadonnées générales pour la description d'un corpus

Dans le but de documenter un corpus partagé, nous proposons d'utiliser un ensemble de métadonnées (cf. Figure 36). Ces métadonnées offrent aux chercheurs intéressés des informations pertinentes sur le contenu du corpus. Elles peuvent aussi être utilisées comme caractéristiques d'une requête lors de l'interrogation de la base des corpus partagés. Une partie de cet ensemble de métadonnées est issue du standard Dublin Core Metadata Initiative (DCMI) (DCMI, 1994). DCMI propose un ensemble de métadonnées génériques définies pour décrire les ressources numériques ou physiques et est un standard de métadonnées indépendant du domaine. À côté des éléments choisis dans le standard DCMI, nous proposons d'ajouter d'autres éléments significatifs pour la description des corpus de traces d'interaction d'apprentissage. Les éléments des métadonnées supplémentaires, sauf l'élément « question de recherche » (cf. ci-après), sont exclusivement utilisés dans la description des corpus initiaux en raison de leur relation avec la situation d'apprentissage tracée dans le corpus.

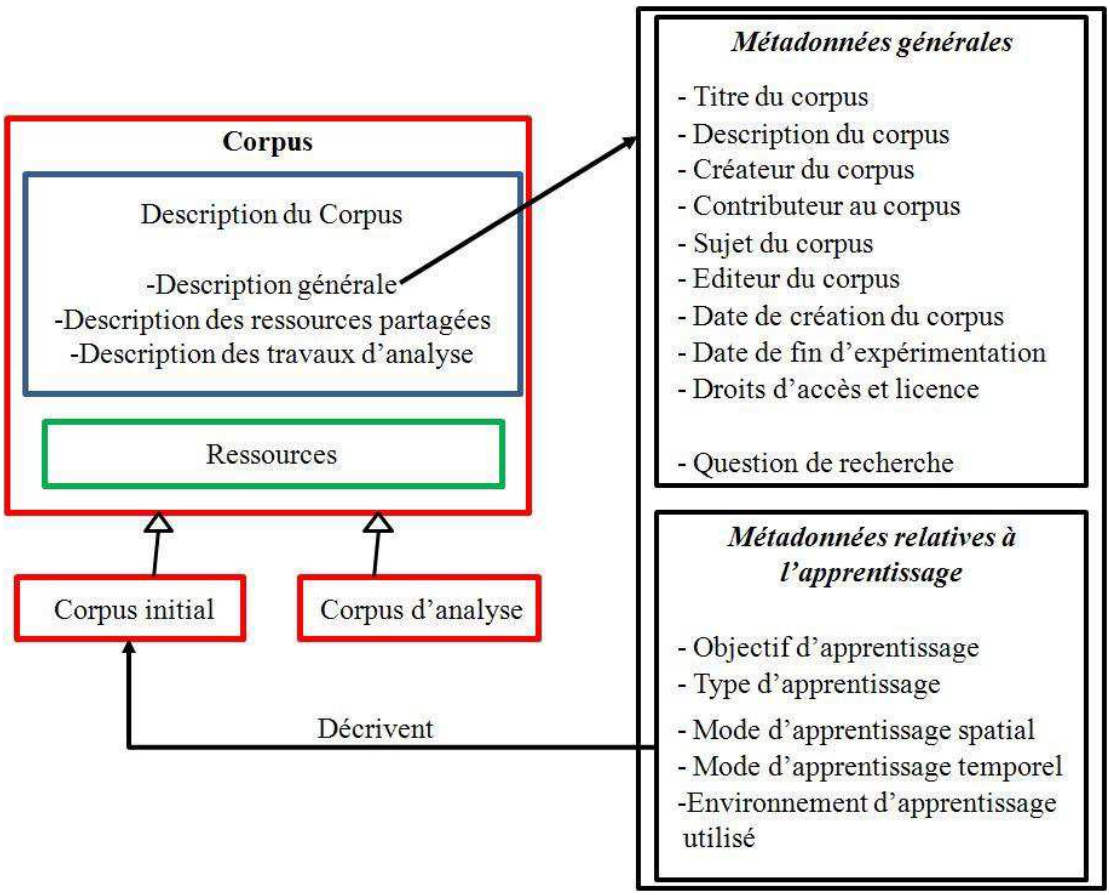


Figure 36 Métadonnées générales pour la description d'un corpus

Nous avons modélisé le corpus à l'aide du langage d'ontologie web OWL (OWL, 2004). Un corpus est représenté par le concept (classe en OWL) « corpus ». Les concepts « corpus initial » et « corpus d'analyse » sont définis comme des sous-concepts de « corpus ». Les métadonnées communes à la description des deux types de corpus sont définies sous formes de propriétés OWL décrivant le concept « corpus ». Les métadonnées spécifiques aux corpus initiaux sont définies sous formes de propriétés OWL décrivant le concept « corpus initial ». Nous nous sommes inspirés dans le choix des métadonnées de description d'un corpus du travail approfondi réalisé dans le cadre du projet MULCE (Chanier et al., 2010).

Les métadonnées provenant du standard DCMI sont ainsi interprétées :

- Titre du corpus : nom donné au corpus, choisi par le chercheur, et qui peut représenter un nom de projet ou d'expérimentation. Cet élément correspond à l'élément « title » des métadonnées du Dublin Core.
- Description du corpus : correspond à l'élément « description » du Dublin Core, cet élément est utile pour donner à un éventuel utilisateur ultérieur du corpus (1) une description sommaire de l'expérimentation tracée dans le corpus (cas d'un corpus initial), ou (2) une description du contenu d'un corpus d'analyse. La description peut être saisie par le chercheur au moment de la création du corpus dans la plate-forme. Sinon, il est possible que le chercheur dispose d'une ou plusieurs ressources de type documentation ou publication. Dans ce cas, une telle ressource est importée dans le corpus et référencée dans cet élément de description.
- Créateur du corpus : cet élément désigne une entité ayant créé le corpus (personne ou organisation), il est possible d'avoir plusieurs créateurs. Cet élément permet de faire référence à un agent ayant participé à la création du corpus (les agents ayant participé à la création ou l'enrichissement d'un corpus ou de ses ressources, ou ayant une relation avec un environnement d'apprentissage utilisé ou un outil d'analyse sont référencés une seule fois dans la plate-forme. Ainsi, si un agent intervient dans plusieurs corpus, environnements d'apprentissage ou outils d'analyse, il est décrit une seule fois). Cet élément correspond à l'élément « creator » du Dublin Core (un agent incarne le concept « Agent » du Dublin Core auquel nous ajoutons les propriétés : nom, institution, courrier électronique et statut).
- Contributeur au corpus : cet élément désigne une entité ayant contribué au corpus, il est possible d'avoir plusieurs contributeurs. Cet élément correspond à l'élément « contributor » du Dublin Core, un contributeur est une personne ou organisation

ayant apporté des contributions au corpus (un contributeur est une entité participante au corpus mais avec moins d'importance qu'un créateur). Cet élément fait référence à un agent ayant contribué au corpus. Un chercheur ayant réalisé un travail d'analyse sur le corpus dont les ressources sont intégrées au corpus doit être ajouté aux contributeurs du corpus.

- Sujet du corpus : cet élément permet de documenter le thème du contenu du corpus et s'exprime sous forme de mots clés ou phrases clés décrivant d'une manière succincte le corpus. Cet élément correspond à l'élément « subject » du Dublin Core.
- L'éditeur du corpus : cet élément, qui correspond à l'élément « publisher » du Dublin Core, désigne l'entité responsable de la mise à disposition du corpus.
- La date de création du corpus : cet élément correspond à l'élément « available » du Dublin Core, et correspond à la date de création du corpus dans la plate-forme de partage pour le mettre à disposition d'autres chercheurs.
- Date de fin de l'expérimentation ayant produit le corpus : cet élément correspond à l'élément « created » du Dublin Core, et correspond à la date première de collecte des ressources du corpus, donc à la date de fin de l'expérimentation. On peut imaginer qu'un corpus collecté il y a une dizaine d'année (date de collecte) peut être reconstruit dans la plate-forme cette année (date de mise à disposition). Nous considérons cette donnée utile pour un chercheur car elle lui permet d'avoir une idée sur la période réelle de l'expérimentation.
- Les droits d'accès et licence associés au corpus : cet élément peut désigner des informations sur les différents droits de propriété sur le corpus et/ou une licence définie par les créateurs du corpus, ces informations peuvent être sous forme de texte saisi par un chercheur à la création du corpus ou de ressource définissant les permissions sur le corpus (exemple : réutilisation, modification, distribution, etc.), il correspond à l'élément « rights » du Dublin Core.

Nous proposons l'ajout de l'élément « question de recherche » associé à tout corpus partagé. Cet élément est important dans la description d'un corpus. Un corpus étant partagé par et pour des chercheurs, ces derniers peuvent être intéressés à connaître les questions de recherche ayant motivé (1) l'expérimentation qui a donné lieu à un corpus initial, ou (2) les travaux d'analyse constituant un corpus d'analyse (un corpus d'analyse étant construit pour étudier une question de recherche particulière). Actuellement, nous considérons que la question de recherche est exprimée sous une forme textuelle informelle compréhensible par le chercheur

qui consulte le corpus ou, le cas échéant, interrogeable par une recherche textuelle par mots clés. Ces éléments de métadonnées permettent la description des deux types de corpus.
Les éléments complémentaires que nous proposons d'ajouter pour la description d'un corpus initial sont :

- Objectif d'apprentissage ayant motivé l'expérimentation : cet élément permet d'informer un chercheur qui explore le corpus sur les objectifs de l'apprentissage de l'expérimentation tracée. Cette information peut aider le chercheur à savoir rapidement si le corpus l'intéresse ou pas. Les objectifs d'apprentissage peuvent être énoncés sous forme de liste d'objectifs par le chercheur au moment de la création du corpus ou importer une ressource, si elle existe, dans laquelle il énonce ces objectifs.
- Type d'apprentissage : cet élément indique si le type d'apprentissage tracé est l'apprentissage *individuel*, l'apprentissage *collectif* ou les deux. En effet, des chercheurs en E.I.A.H. peuvent s'intéresser plus particulièrement à l'apprentissage individuel ou collectif, et non aux deux en même temps.
- Aspect spatial d'apprentissage : cet élément donne au chercheur une idée sur le mode d'apprentissage choisi dans l'expérimentation par rapport à l'espace, qui peut être un apprentissage en *présentiel*, à *distance* ou mixte.
- Aspect temporel d'apprentissage : cet élément renseigne le chercheur sur le mode d'apprentissage par rapport au temps, si les activités d'apprentissage proposées aux apprenants sont *synchrones* ou *asynchrones*, ou les deux.
- Environnement d'apprentissage utilisé : cet élément référence un outil d'apprentissage utilisé dans l'expérimentation d'apprentissage. Comme pour un agent ayant créé ou contribué à la création d'un corpus, les outils d'apprentissage sont décrits en dehors du corpus puisqu'un outil d'apprentissage peut être utilisé dans différentes expérimentations et peut donc être référencé dans différents corpus. Cet élément peut être intéressant pour un chercheur souhaitant étudier les traces générées par un outil d'apprentissage particulier ou par un type particulier d'outils d'apprentissage.

4.3.5.2 Métadonnées de description des ressources partagées dans un corpus

Comme déjà présenté dans le paragraphe 4.3.3, des ressources de différents types peuvent être partagées dans un corpus. Afin de permettre au chercheur de bien comprendre la structure du corpus et les types et contenus des ressources, des métadonnées sont fournies

pour décrire les ressources partagées dans un corpus (cf. Figure 37). Les métadonnées sont également utiles dans l'interrogation des corpus. Comme pour la description d'un corpus, nous réutilisons une partie des métadonnées du standard Dublin Core auxquelles nous ajoutons des métadonnées supplémentaires suivant les besoins de description qui peuvent varier selon le type de la ressource décrite.

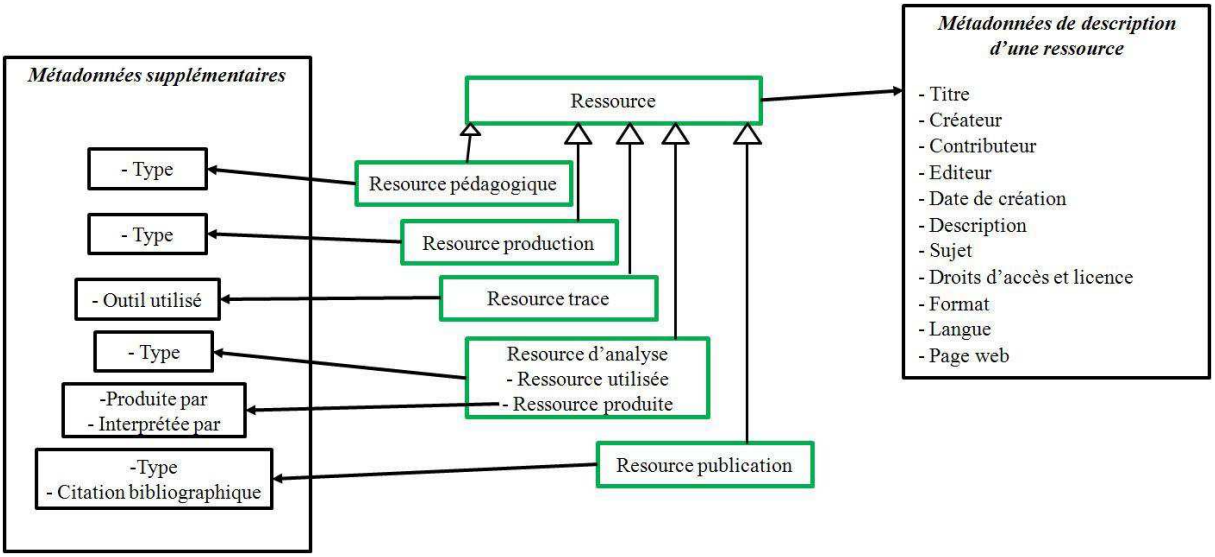


Figure 37 Métadonnées de description des ressources

Les métadonnées du Dublin Core que nous réutilisons et qui décrivent tous les types de ressources sont :

- Titre de la ressource : nom donné à la ressource partagée, (élément « title » du Dublin Core).
- Créateur de la ressource : comme pour un corpus, un créateur est un agent ayant participé à la création de la ressource (élément « creator » du Dublin Core). Une ressource peut avoir plusieurs créateurs.
- Contributeur : agent ayant contribué au contenu du corpus de manière moins importante qu'un créateur (élément « contributor » du Dublin Core). Une ressource peut avoir plusieurs contributeurs.
- Editeur : agent responsable de la mise à disposition de la ressource (élément « publisher » du Dublin Core).
- Date de création de la ressource (élément « created » du Dublin Core).
- Description de la ressource : permet d'avoir une idée sur le contenu de la ressource (élément « description » du Dublin Core).

- Sujet de la ressource : un ensemble de mots clés et de phrases clés permettant une description brève du contenu de la ressource (élément « subject » du Dublin Core).
- Droits d'accès et licence : désigne les droits détenus sur la ressource et/ou la licence (donnant des permissions sur la ressource) associés à la ressource décrite (élément « rights » du Dublin Core).
- Format de la ressource : correspond à l'élément « format » du Dublin Core, cet élément est choisi parmi les types du standard Multipurpose Internet Mail Extensions (MIME).
- Langue de la ressource : la langue des données contenues dans la ressource, correspond à l'élément « language » du Dublin Core, les langues sont représentées selon la norme RFC-5646 (RFC-5646, 2009) d'identification des langues.

Par ailleurs, nous définissons d'autres métadonnées pour la description des ressources partagées dans un corpus. Pour toute ressource partagée, si celle-ci est accessible sur Internet, une métadonnée « page web » est associé à la ressource. Ceci peut être utilisé pour accéder à des ressources de documentation éparpillées (par exemple une page web dans laquelle un chercheur décrit une expérimentation). Ici, deux types de problèmes peuvent être soulevés : (1) la pérennité des ressources Web, en effet, les pages Web peuvent devenir obsolète après un certain temps pour diverses raisons (p. ex. fin d'un projet, migration des données, etc.), et (2) la modification du contenu des ressources (le contenu d'une page web peut évoluer dans le temps ce qui fait que la ressource devient différente de celle référencée initialement). Une solution à ces deux problèmes peut être de créer une sorte de service d'archive qui garde des *snapshots* des ressources Web indexées par la date un peu à l'image des archives internet (archive.org, 2013).

La métadonnée « type » du standard Dublin Core est utilisée pour typer une ressource, nous la définissons pour les types des ressources suivants :

- ressources pédagogiques (les types possibles proposés, pouvant être enrichis, sont représentés sur la Figure 31),
- ressources productions (cf. Figure 32),
- ressources d'analyse utilisées (cf. Figure 33),
- ressources d'analyse produites (cf. Figure 34),
- et ressources de publication (cf. Figure 35).

La métadonnée « citation bibliographique » du standard Dublin Core est utilisée dans la description d'une ressource de type publication, elle permet de référencer la communication dans des travaux ultérieurs et de la retrouver facilement.
Deux métadonnées supplémentaires sont proposées pour décrire une ressource d'analyse produite ; la première « produite par » et qui référence l'outil d'analyse utilisé dans l'analyse ayant produit la ressource décrite, et la deuxième « interprétée par » liant la ressource décrite à une éventuelle ressource d'interprétation qui l'interprète. Une telle ressource peut faire l'objet de nouvelles analyses cumulatives auquel cas, la structure des données qu'elle contient doit être définie (sous forme d'un schéma XML par exemple accompagné éventuellement d'une description de la structure) et liée à la ressource. Ce type de données ne fait pas vraiment partie des métadonnées descriptives de la ressource mais est nécessaire pour son interrogation ultérieure.

Enfin, une ressource de type trace est décrite par une métadonnée supplémentaire « outil utilisé », qui fait référence à un outil offert aux apprenants par la plate-forme d'apprentissage durant l'expérimentation. L'objectif est de savoir quels outils, parmi ceux utilisés par l'expérimentation, sont tracés dans une ressource trace particulière. Ceci peut être utile lorsque l'on est intéressé par l'analyse de traces d'utilisation d'un outil particulier. Une ressource trace directement traitable par une machine est liée à une ressource définissant la structure des données de la trace, et munie idéalement d'une description de cette structure permettant d'expliciter les données contenues dans une ressource trace.

4.3.5.3 Modèle de processus d'un travail d'analyse

Nous proposons un modèle de processus d'analyse des corpus partagés par la plate-forme et utilisant les outils d'analyse partagés. Ce modèle est illustré dans la Figure 38 ci-dessous. Un chercheur, souhaitant étudier une question de recherche particulière, se fixe des objectifs d'analyse pour le travail d'analyse à réaliser. Ayant accès à la base de corpus partagés, le chercheur cherche dans les corpus d'analyse existants, si un corpus d'analyse avec sa même question de recherche existe déjà. Une telle recherche peut être réalisée via un système de recherche par mots clés. Si un tel corpus d'analyse existe, le chercheur peut le choisir pour contenir son travail d'analyse. Sinon, le chercheur construit un nouveau corpus d'analyse pour la question de recherche étudiée. Un travail d'analyse peut être réalisé à l'aide d'un ou plusieurs outils d'analyse. Le chercheur doit donc choisir parmi les outils d'analyse offerts par la plate-forme, le ou les outils utiles pour son travail d'analyse. Pour chaque outil

d'analyse, le chercheur choisit le ou les corpus sources, c'est-à-dire le ou les corpus initiaux et/ou d'analyse dont les ressources serviront à l'analyse. Pour chacun des corpus choisis, le chercheur désigne les ressources du corpus, qui serviront à ses analyses. De telles ressources peuvent être :

- des ressources traces (une ressource de type trace existe obligatoirement dans un corpus original),
- des ressources traces enrichies par des analyses (ressources produites par l'analyse),
- des ressources contextuelles utilisées par l'analyse contenant des données intéressantes pour l'analyse (par exemple des données concernant les participants à l'apprentissage),
- des ressources utilisées par des analyses antérieures que le chercheur souhaite réutiliser (par exemple un schéma de codage).

Pour chaque ressource désignée, deux cas se présentent : (1) la ressource est examinée et les données nécessaires pour l'analyse sont extraites et converties au format d'entrée de l'outil d'analyse ; (2) la ressource est importée telle quelle si une conversion n'est pas nécessaire.

Une fois que toutes les ressources des corpus sources désignées sont examinées, et que les données extraites, converties et formatées sont importées dans l'outil d'analyse, le chercheur a la possibilité d'importer des ressources complémentaires qui lui seront utiles dans son analyse (par exemple un modèle de définition des connaissances d'un domaine d'application) avant de commencer son analyse avec l'outil d'analyse. Le travail réalisé avec l'outil d'analyse produit des ressources (*ressource d'analyse produite*) qui seront intégrées au corpus d'analyse contenant le travail en cours. Les résultats enregistrés dans les ressources résultantes de l'utilisation d'un outil d'analyse peuvent être interprétés. Si ces interprétations sont documentées, elles doivent être intégrées au corpus d'analyse en tant que ressources d'interprétation.

Enfin, une activité ne faisant pas partie du processus d'un travail d'analyse mais qui lui est complémentaire, est celle de la communication scientifique par la publication d'articles liés au travail d'analyse. Ces communications doivent être intégrées au corpus d'analyse en tant que ressources de type publication et liées aux travaux d'analyse qu'elles décrivent.

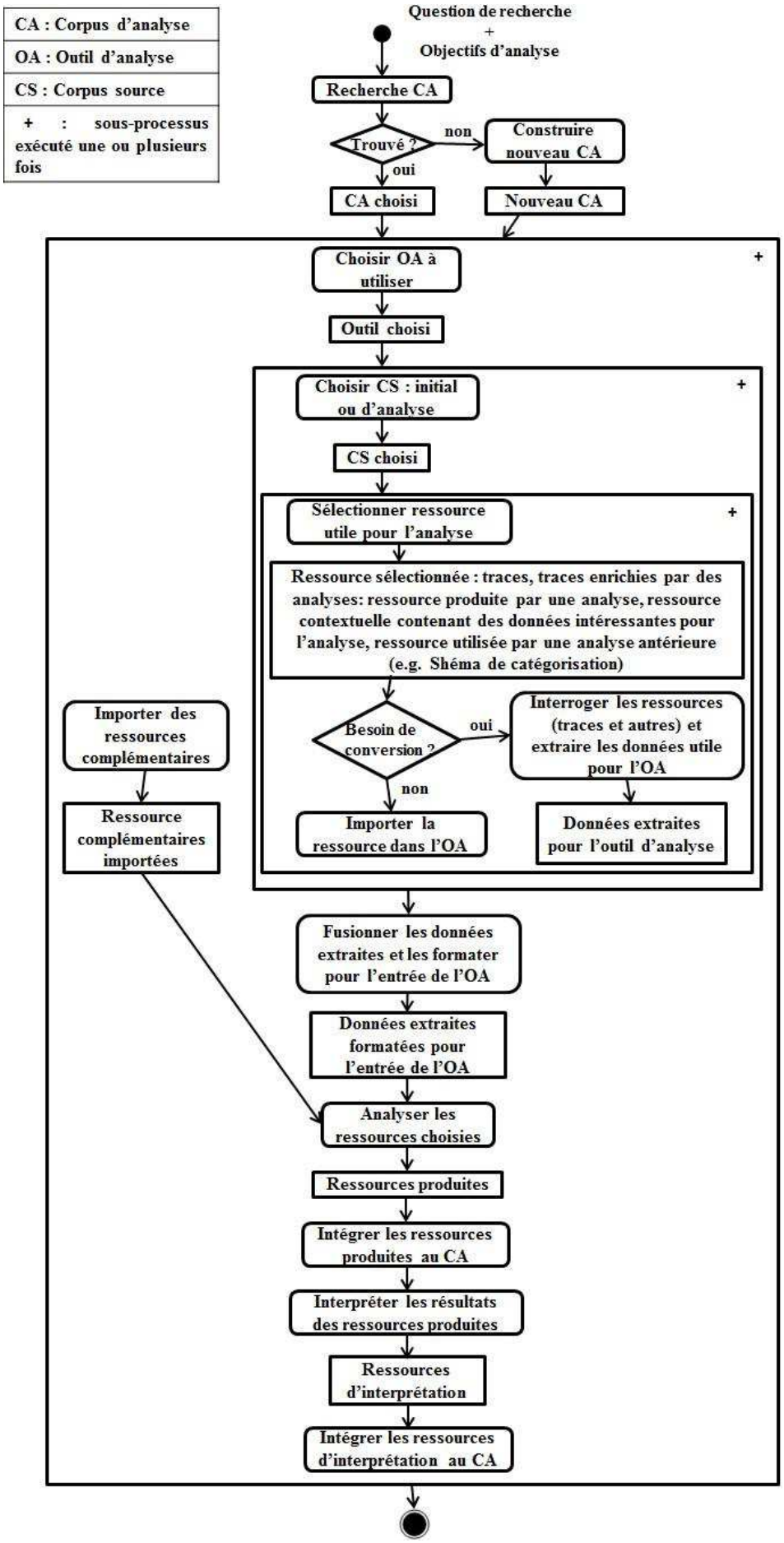

CA : Corpus d'analyse
OA : Outil d'analyse
CS : Corpus source
+ : sous-processus exécuté une ou plusieurs fois
Question de recherche
+
Objectifs d'analyse
Recherche CA
Trouvé ?
non
Construire nouveau CA
oui
CA choisi
Nouveau CA
Choisir OA à utiliser
+
Outil choisi
Choisir CS : initial ou d'analyse
+
CS choisi
Sélectionner ressource utile pour l'analyse
+
Ressource sélectionnée : traces, traces enrichies par des analyses: ressource produite par une analyse, ressource contextuelle contenant des données intéressantes pour l'analyse, ressource utilisée par une analyse antérieure (e.g. Shéma de catégorisation)
Besoin de conversion ?
oui
Interroger les ressources (traces et autres) et extraire les données utile pour l'OA
non
Importer la ressource dans l'OA
Données extraites pour l'outil d'analyse
Importer des ressources complémentaires
Ressource complémentaires importées
Fusionner les données extraites et les formater pour l'entrée de l'OA
Données extraites formatées pour l'entrée de l'OA
Analyser les ressources choisies
Ressources produites
Intégrer les ressources produites au CA
Interpréter les résultats des ressources produites
Ressources d'interprétation
Intégrer les ressources d'interprétation au CA

Figure 38 Modèle du processus d'un travail d'analyse

4.3.5.4 Description des travaux d'analyse contenus dans un corpus

La description d'un travail d'analyse réalisé à l'aide d'un ou plusieurs outils sur un ensemble de corpus initiaux ou d'analyses, permet de garder des données sur le travail d'analyse. Ces données sont utiles dans la consultation d'un corpus. Ainsi, un chercheur souhaitant explorer les corpus partagés dans la plate-forme a la possibilité de consulter une description des travaux d'analyse contenus dans le corpus. Une description donne des informations générales sous forme de métadonnées. Elle permet également de garder la trace des outils utilisés et des différentes ressources ayant servies dans l'analyse ainsi que les scripts qui ont permis l'extraction des données, leur conversion et leur formatage. Ceci permet non seulement de consulter une analyse réalisée, mais aussi de la reproduire et la vérifier, ou encore de la contredire en travaillant sur les mêmes données et en publiant des résultats différents. Par ailleurs, il est également possible de capitaliser sur des analyses antérieures en travaillant sur les ressources qu'elles produisent. La description d'un travail d'analyse doit donc permettre de garder le maximum d'informations pour le documenter indépendamment des différentes ressources pouvant être utilisées et produites durant le processus et qui peuvent avoir des contenus sémantiques très différents d'une analyse à une autre ainsi que des formats de représentation différents.

Comme déjà mentionné, nous considérons la possibilité qu'un chercheur qui partage un corpus ait déjà réalisé un ou plusieurs travaux d'analyse sur ce corpus. Ces analyses réalisées hors de la plate-forme de partage sont tout aussi intéressantes à décrire. Il est également possible d'envisager le cas d'analyses réalisées en dehors de la plate-forme après le partage du corpus. Toute analyse réalisée sur un corpus, au sein ou en dehors de la plate-forme, doit être décrite, et le maximum de données et ressources de documentation doivent être partagées. Cela permet de capitaliser les travaux des chercheurs qui peuvent être très variés, permettant ainsi un large partage au sein de la communauté des chercheurs. Nous proposons de décrire un travail d'analyse (cf. Figure 39) par :

- Date de début : date à laquelle le travail d'analyse a débuté.
- Date de fin : date de fin de l'analyse.

- Description : élément « description » du standard Dublin Core, désigne une description textuelle du travail d'analyse. Peut être saisie par le chercheur décrivant l'analyse ou être exprimée dans une ressource de documentation ou de publication que le chercheur importe et référencie.
- Créateur : agent responsable de la réalisation du travail d'analyse. Il est possible qu'un travail d'analyse ait plusieurs créateurs. Cette métadonnée correspond à l'élément « creator » du standard Dublin Core.
- Contributeur : agent ayant contribué à la réalisation du travail d'analyse avec une moindre importance qu'un créateur. Plusieurs contributeurs peuvent être associés à un travail d'analyse. Cette métadonnée correspond à l'élément « contributor » du standard Dublin Core.
- Objectif d'analyse : décrire un objectif d'analyse que le chercheur se fixe pour son travail d'analyse. Plusieurs objectifs peuvent être fixés pour un travail d'analyse. Cet élément permet de définir différents objectifs pour étudier la question de recherche ayant fait l'objet du corpus d'analyse. Si le chercheur définit une ou plusieurs ressources explicitant les objectifs d'analyse, ces ressources peuvent être importées sous forme de ressources de documentation et référencées.

Un travail d'analyse étant de nature complexe, nous considérons la possibilité qu'un chercheur s'intéresse à l'analyse de plusieurs corpus en utilisant un ou plusieurs outils d'analyse. Ainsi, une « analyse par corpus » est décrite par :

- Référence à l'outil d'analyse utilisé dans la réalisation d'une partie du travail d'analyse décrit. Les outils d'analyse utilisés par les chercheurs, qu'ils soient partagés dans la plate-forme ou non, sont décrits d'une manière centralisée permettant d'éviter de reprendre la description de l'outil. La description d'un outil d'analyse sera présentée dans la suite de ce chapitre.
- Créateur : référence au(x) créateur(s), parmi les agents définis comme créateurs du travail d'analyse (cf. ci-dessus), ayant créé une partie du travail d'analyse réalisée avec un outil particulier (élément « creator » du Dublin Core).
- Contributeur : référence au(x) contributeur(s), parmi les agents définis comme contributeurs du travail d'analyse (cf. ci-dessus), ayant contribué à une partie du travail d'analyse réalisée avec un outil particulier (élément « contributor » du Dublin Core).

- Description de l'analyse réalisée avec un outil d'analyse. Cet élément, s'il est fourni, peut décrire davantage le travail particulier réalisé avec un outil d'analyse dans le cadre d'un travail d'analyse. La description peut être textuelle saisie ou être exprimée dans une ressource de documentation ou de publication qui est importée dans le corpus et référencée (élément « description » du Dublin Core).
- Date d'extraction des données, provenant des corpus partagés, et analysées par l'outil d'analyse particulier permettant la réalisation d'une partie d'un travail d'analyse décrit. Cette donnée assure la reproductibilité du résultat d'analyse, et permet de récupérer les mêmes données d'un corpus interrogé même si celui-ci a été enrichi par ailleurs.
- Références aux ressources interrogées du corpus et aux scripts utilisés dans l'extraction, le filtrage et le formatage des données de ces ressources. L'interrogation de différents corpus et des ressources de ces corpus se fait à l'aide de différents scripts permettant l'extraction des données utiles pour l'analyse en considérant la différence des formats de représentation de ces ressources. Nous avons défini différents types de scripts relatifs aux opérations d'extraction, de conversion, de filtrage et de formatage des données. Ces différentes opérations seront présentées dans le chapitre 6. Dans la description d'une « analyse par corpus », il convient de préciser pour chaque ressource utilisée de ce corpus (ou ensemble de ressources si elles ont la même structure) l'ensemble des scripts utilisés pour préparer les données de la ressource à l'entrée de l'outil d'analyse. Ceci permet de garder le détail des données ayant servi pour l'analyse en donnant la possibilité de ré-exécuter les scripts permettant la préparation de l'entrée de l'outil d'analyse.
- Références à des éventuelles ressources complémentaires importées et utilisées dans l'analyse (ressources d'analyse utilisées). Ces ressources n'existent pas dans les corpus interrogés mais sont utilisées par le chercheur durant son travail en fonction de ses besoins d'analyse. Cette information permet de garder une trace de toute ressource ayant servi durant l'analyse, et donne la possibilité à d'autres chercheurs de reproduire l'analyse.
- Références à des ressources produites (ressources d'analyse produites) suite à l'utilisation de l'outil d'analyse. De telles ressources permettent à un autre chercheur souhaitant reproduire l'analyse de comparer ses résultats à ceux publiés, ou encore de réaliser des analyses cumulatives sur les résultats déjà obtenus (à condition que ces

ressources aient des structures exploitables pour être interrogées, p. ex. un fichier XML, CSV, etc.).

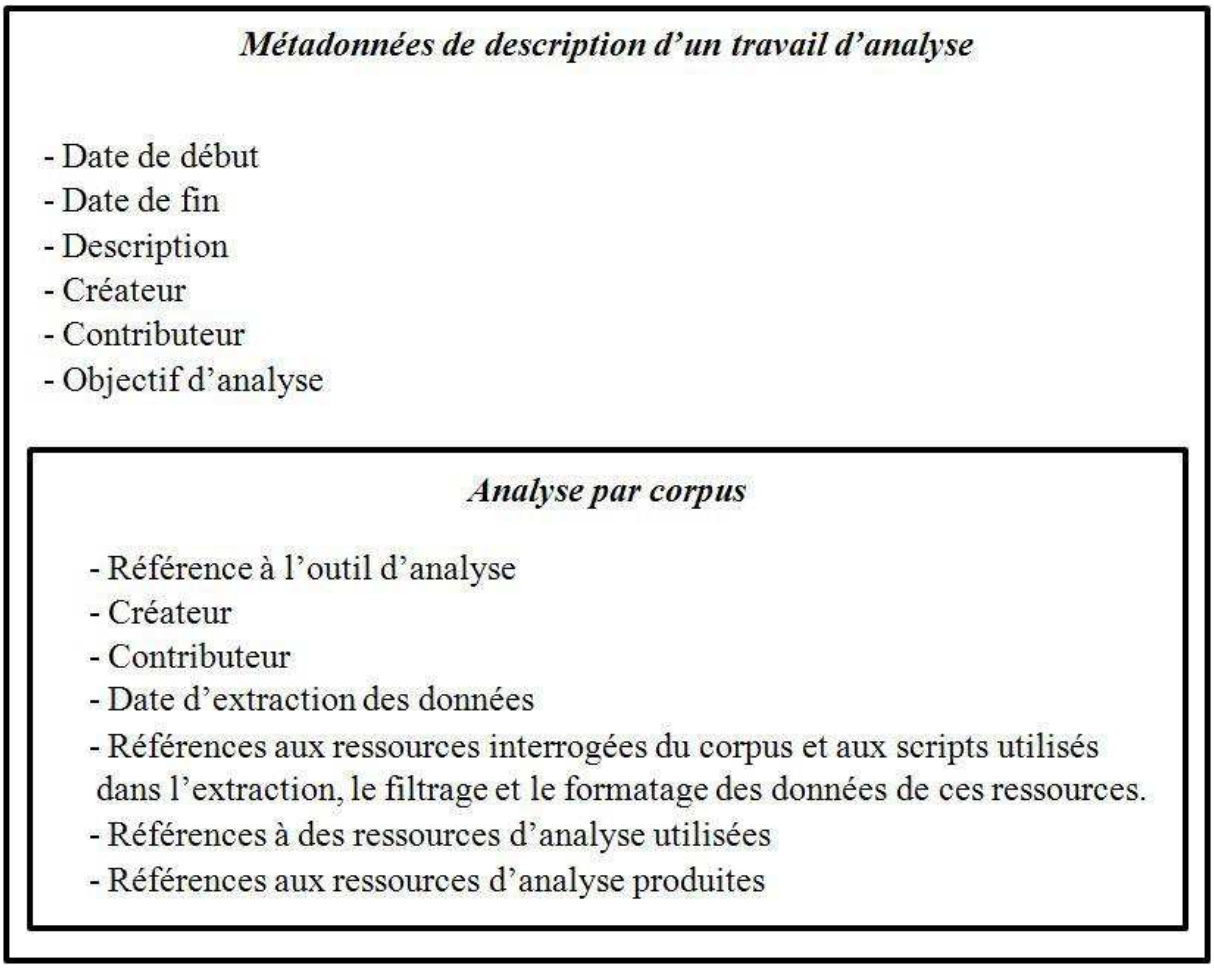


Figure 39 Description d'un travail d'analyse

4.4 Synthèse

Le modèle de description du corpus permet d'apporter des réponses aux contraintes énoncées dans la section 3.3, qui concernent l'hétérogénéité des traces, la différence de nature des traces, et le besoin de contextualisation des traces d'interaction. En effet, notre approche ayant fait le choix de ne pas imposer une représentation commune des données, un corpus ne nécessite pas de conversions préalables et contient les ressources hétérogènes collectées par le chercheur permettant ainsi de garder les données initiales et leur richesse sémantique. Ces ressources peuvent contenir non seulement des traces d'interaction, mais aussi des données contextuelles. Par ailleurs, la description d'un travail d'analyse contribue à la capitalisation des analyses en gardant la trace des analyses antérieures réalisées sur les corpus partagés. Les

métadonnées que nous utilisons dans la description d'un corpus et de ses composants contiennent un sous-ensemble de métadonnées génériques réutilisées du standard de métadonnées DCMI donnant des informations générales de documentation sur les objets décrits, et des métadonnées supplémentaires que nous définissons pour :

- rendre compte de l'aspect apprentissage des corpus partagés,
- garder des données pouvant servir dans l'interrogation des ressources partagées (p. ex. quels outils sont tracés dans une ressource trace, quelle ressource contient l'interprétation d'une ressource produite par une analyse, etc.),
- garder des données permettant la reproduction des analyses antérieures en précisant les ressources interrogées et les scripts utilisés par l'interrogation.

En proposant ce modèle de corpus, nous avons fait le choix de ne pas définir de structures pour les ressources composant les corpus. L'approche consiste à (1) fournir des métadonnées générales de description du corpus, (2) collecter, classifier, et décrire les ressources (traces et autres) fournies par les chercheurs, et (3) décrire les analyses réalisées et les lier aux corpus. Tous les travaux étudiés dans l'état de l'art utilisent des formats spécifiques dans la collecte de leurs corpus ce qui peut causer des pertes sémantiques sur les contenus initiaux des corpus bruts. Les projets CALICO (Giguet el al., 2009), dataTel (Drachsler et al., 2010), et CAM-CIM (Butoianu et al., 2010), bien qu'ayant énoncé le partage de corpus comme l'un des objectifs de leur travaux, ne traitent pas l'aspect contextualisation des traces collectées. Ils ne s'intéressent qu'aux traces, et n'abordent pas le partage de données et de ressources contextuelles permettant d'expliciter les données partagées. Les projets Datashop (Koedinger et al. 2008) et Undertracks (Bouhineau et al., 2013a) proposent la contextualisation des traces au moyen de ressources de description pour le premier et par le partage d'articles scientifiques liés au corpus pour le second. Le projet MULCE (Reffay et al., 2008) s'est par contre attardé beaucoup plus sur la formalisation du contexte. Par exemple, ce projet préconise la formalisation des scénarios pédagogiques, élément essentiel du composant contexte selon ce projet, en utilisant le langage IMS-LD (IMS-LD, 2003) de scénarisation pédagogique. En ce qui concerne notre travail, nous proposons une solution intermédiaire. En effet, nous récupérons toute ressource fournie telle quelle par le chercheur et qui peut s'avérer utile dans la documentation d'un corpus, mais en essayant de classifier ces ressources de manière thématique (pédagogie, production, publication, etc.) et en les décrivant avec des métadonnées. Ces métadonnées de description permettent une recherche facile dans la base de corpus mais ne nécessitent aucune structuration profonde de ces ressources. Si un scénario

pédagogique est formalisé en utilisant le standard IMS-LD, cela le rendra éventuellement plus réutilisable, sinon avoir un scénario pédagogique exprimé dans un autre langage (p. ex. LDL (Martel et al., 2006)) ou simplement exprimé sous forme textuelle demeure intéressant pour le partage. Par ailleurs, des standards de structuration de ressources pédagogiques tels que IMS Content Packaging (IMS-CP, 2009) sont utilisés pour permettre l'interopérabilité des ressources entre différentes plateformes d'apprentissage. Bien que les langages standards soient de plus en plus utilisés, ils ne sont pas encore généralisés et la définition formelle d'un scénario pédagogique n'est pas toujours une étape indispensable. C'est pourquoi nous avons fait le choix de collecter les ressources disponibles et d'essayer de les décrire avec un ensemble de métadonnées. Nous nous sommes inspirés du travail réalisé dans le cadre du projet MULCE (Chanier et al. 2010) en utilisant un ensemble de métadonnées génériques complétées par des métadonnées liées au domaine de l'apprentissage.

Il est important de noter que la disponibilité de ressources contextuelles n'est pas évidente car les chercheurs qui montent une expérimentation n'ont pas forcément l'intention de partager leurs corpus. Le modèle de corpus que nous proposons permettrait de récupérer autant que possible le maximum de ressources contextuelles. Ce modèle, couplé à la plateforme de partage pourrait encourager les chercheurs à mieux documenter leurs corpus pour faciliter le partage.

Le modèle de corpus présenté dans ce chapitre n'imposant pas de représentations particulières des ressources partagées dans un corpus, permet de représenter différents corpus hétérogènes d'une manière générique consistant à accepter les ressources dans leurs formats originaux et de définir une description unique du corpus et de ses composants. Ce modèle rend notre approche flexible puisqu'il ne contraint pas les chercheurs à formaliser leurs données en utilisant une représentation particulière. Enfin le modèle de corpus proposé garantit l'utilisabilité en définissant un ensemble minimal de métadonnées et différents types de ressources contextuelles permettant de renseigner le chercheur sur le contenu d'un corpus.

Chapitre 5 : L'approche « Proxyma » : modèle Sémantique des concepts interrogeables

5.1 Introduction

Ce chapitre présente le deuxième modèle de l'approche « Proxyma » : le modèle sémantique. Ce dernier permet de définir des concepts consultables, aidant les chercheurs à analyser les corpus partagés. Ce deuxième modèle de l'approche « Proxyma » représente l'alternative que nous proposons pour éviter d'imposer un nouveau format pour la représentation des traces d'interaction. C'est en se basant sur ce modèle qu'un chercheur peut réaliser le travail d'alignement entre les concepts de la taxonomie, définie dans l'ontologie, relatifs à ce modèle et leur incarnation dans un corpus donné. Ce travail est fondamental du

point de vue de l'approche « Proxyma » car il permet l'interrogation ultérieure des ressources traces du corpus afin de les analyser. L'approche « Proxyma » étant incrémentale et participative, ce modèle peut évoluer pour permettre aux chercheurs d'ajouter des concepts spécifiques plus élaborés leur permettant d'élargir le spectre des analyses réalisables sur les corpus. Nous avons choisi de conceptualiser les trois modèles de l'approche « Proxyma », dont le modèle sémantique faisant l'objet du présent chapitre, en utilisant une ontologie descriptive. Le modèle sémantique correspond à une taxonomie des concepts interrogeables des corpus. Dans la suite, nous ferons référence à ces concepts interrogeables comme faisant partie de la taxonomie ou de l'ontologie indifféremment.

Nous commençons par la présentation du principe général du modèle. Nous introduisons ensuite la genèse des concepts du modèle sémantique pouvant peupler la taxonomie. Nous définissons par la suite les relations pouvant exister entre les concepts définis dans le modèle, ce qui permet de distinguer deux types de concepts. En effet, une taxonomie définit généralement une relation hiérarchique entre les concepts qui la composent. Nous enrichissons la notion de taxonomie en définissant un deuxième type de relation entre les concepts. Enfin, nous présentons un ensemble de catégories de concepts que l'on a distinguées d'une manière thématique pour recenser quelques concepts pouvant appartenir au modèle sémantique.

5.2 Principe du modèle

Ce modèle concerne la sémantique des données collectées dans un corpus de traces d'interaction et dont l'interrogation est nécessaire pour une analyse à l'aide d'un outil d'analyse partagé dans la plate-forme. L'interrogation concerne essentiellement les ressources traces contenues dans un corpus. Un corpus peut contenir des ressources d'analyse produites suite à l'analyse de ressources traces. Ainsi, une ressource d'analyse peut dans certains cas être vue comme une ressource trace enrichie par des analyses. De telles ressources peuvent à leur tour être utilisées dans le cadre d'analyses cumulatives. Par ailleurs, dans son analyse des traces, un chercheur peut s'intéresser à des ressources contextuelles contenant des données pertinentes pour l'analyse mais qui n'existent pas dans la trace. Le modèle sémantique définit donc des concepts présents dans les ressources traces, les ressources d'analyse produites (traces enrichies par des analyses), et les ressources utilisées durant l'analyse.

Un problème se pose à ce niveau, car les frontières entre les concepts provenant de ces trois types de ressources ne sont pas facilement définissables. En effet, la différence dans les niveaux de granularité des traces peut faire qu'un même concept peut dans un cas être directement présent dans la trace collectée par un outil d'apprentissage alors qu'il se retrouve, dans un autre cas, dans une ressource trace enrichie par une analyse. Dans ce deuxième cas, l'analyse sert à modifier le niveau de granularité des données pour donner plus de sens à la trace d'interaction première. Par exemple, certains outils tracent des événements de très bas niveau, comme dans Teleos (Luengo et al., 2006), un EIAH en chirurgie orthopédique où des traces d'un dispositif haptique enregistrant les coordonnées de déplacement sont collectées. Ces traces sont utilisées pour diagnostiquer les connaissances de l'apprenant. Par contre, dans le cas de l'EIAH AMBRE-ADD (Duclosson, 2004) d'enseignement d'une méthode de résolution de problèmes additifs exploitant le paradigme de raisonnement à partir de cas, le diagnostic est directement calculé par l'EIAH et est exprimé dans la trace générée par l'outil, qui lui est fortement couplée. Par ailleurs, les données contextuelles (p. ex. métadonnées concernant les apprenants) peuvent faire partie de la ressource de traces ou être exprimées dans une ressource contextuelle (ressources d'analyse utilisée). Nous choisissons de ne pas classer les concepts en fonction des types des ressources qui contiennent les données qui leur sont relatives puisqu'une telle classification risque de ne pas convenir aux différents cas des corpus (p. ex le concept « groupe d'un participant » qui peut être retrouvé dans un fichier de type traces ou encore dans une ressource contextuelle contenant des informations relatives au profil de l'apprenant). Nous faisons donc le choix de classer les concepts par thème ce qui facilite au chercheur la lecture de l'ontologie et l'identification des concepts qui l'intéressent.

Notre objectif étant de proposer une solution flexible et générique, l'idée est de définir un ensemble de concepts, pouvant être liés entre eux, pour servir à l'élaboration d'une sémantique des données contenues dans les corpus commune aux différents chercheurs qui utilisent la plate-forme. Ce modèle sémantique permet, en couplage avec le modèle opérationnel qui sera présenté dans le chapitre suivant, de réaliser l'interopérabilité entre les corpus et les outils d'analyse partagés tout en évitant d'imposer une représentation des données devant être respectée comme condition sine qua non pour cette interopérabilité. Cette approche nous permet donc de pallier le problème de diversité de formats de représentation des données tout en étant plus flexible.

Nous avons antérieurement présenté le concept de « proxy » qui, dans le contexte de notre travail, est utilisé pour illustrer le caractère intermédiaire et transparent des traitements

faits sur les données pour les extraire et les convertir afin de les préparer pour l'entrée d'un outil d'analyse. Il fait donc allusion à l'ensemble de traitements permettant de faire le lien entre les concepts du modèle sémantique défini par la taxonomie et le contenu interrogeable des corpus en utilisant les différentes opérations définies par le modèle opérationnel. Ce dernier définit, nous le verrons, six types d'opérations permettant d'interroger un corpus d'une manière transparente en utilisant les concepts de la taxonomie. Il permet l'extraction des données utiles pour une analyse. Lorsque les scripts relatifs aux opérations nécessaires sont définis pour un couple <<ensemble de concepts intéressants pour un chercheur, corpus à interroger>>, l'interrogation peut se faire d'une manière automatique et transparente. L'idée est de permettre au chercheur de :

- partager toute ressource qu'il juge pertinente pour la contextualisation et l'analyse sans contraintes par rapport à la représentation des données,
- définir des « proxys » permettant l'interrogation de ces corpus afin d'en extraire les données relatives à un sous-ensemble des concepts définis dans l'ontologie.

Un concept défini dans l'ontologie se voit attribuer un nom et une définition textuelle exprimée à un niveau sémantique qui se veut « naturel » pour le chercheur. L'hétérogénéité des données collectées pose le problème de spécificité des concepts et de leur lien étroit avec un domaine d'application particulier, ou plus encore, avec les outils logiciels utilisés, ce qui est contradictoire avec la généricité. Par ailleurs, la différence de niveau de granularité des données pose le problème du niveau sémantique des concepts à définir dans l'ontologie. Dans certains cas, les traces collectées par un outil d'apprentissage particulier doivent obligatoirement subir des transformations préliminaires (processus souvent intégré à l'environnement d'apprentissage lui-même) permettant de les interpréter pour diagnostiquer l'activité afin de suivre et/ou guider l'apprenant dans son apprentissage. Ces analyses préliminaires sont souvent réalisées de manière ad-hoc donnant lieu à des données interprétées de niveau sémantique plus élevé, et pouvant être analysées à leur tour (p. ex. l'EIAH Teleos (Luengo et al., 2006)). Les traces de niveau de granularité bas sont aussi souvent utilisées par des outils de « rejouage » intégrés aux environnements d'apprentissage (p. ex. Aplusix (Nicaud et al., 2002) et Drew (Corbel et al., 2003)). L'hétérogénéité des données et la multitude des domaines d'application adressés par les EIAH font du processus de définition des concepts un processus continu, participatif et incrémental. En effet, l'approche consiste à définir une taxonomie centrale commune définissant des concepts relativement génériques et qui sont, le plus possible, indépendants du domaine d'application

d'un EIAH. Par ailleurs, si un corpus contient des données définissant des concepts utiles pour l'analyse mais qui sont spécifiques à l'outil, ces concepts peuvent être définis sous forme de « module ontologique » spécifique pour l'interrogation des données faisant référence à ces concepts. Un module spécifique est intégré à la taxonomie de base au besoin pour exprimer des concepts supplémentaires. Dans le cas où des chercheurs différents définissent de manière indépendante des modules ontologiques spécifiques, et que ces modules aient un ensemble de concepts en commun, il sera possible d'aligner sémantiquement les concepts des modules différents. Un tel alignement permet une communication sémantique entre les chercheurs définissant et/ou utilisant les concepts, ainsi que la réutilisation des mécanismes d'interrogation définis pour les concepts d'un module. Cet aspect de l'alignement entre modules ontologiques sera abordé dans le chapitre 7.

5.3 Genèse des concepts

Pour mieux comprendre les différentes catégories de concepts définies dans la taxonomie, nous commençons par présenter la genèse des concepts définis. Cette genèse des concepts relatifs au modèle sémantique des concepts interrogeables des corpus est liée à différentes origines. En effet ces concepts peuvent être :

- Des concepts universels qui peuvent se retrouver dans les corpus indépendamment de l'environnement d'apprentissage et du domaine d'application, par exemple, les concepts « participant » et « indicateur temporel » ;

- Des concepts induits par les outils de l'environnement d'apprentissage ; en effet, deux outils de même type peuvent fournir des fonctionnalités différentes ce qui affecte les traces générées. Par ailleurs, les concepteurs qui choisissent les données collectées dans les traces, peuvent se concentrer sur des données différentes. Par exemple, les concepts liés aux interactions dans un forum peuvent différer d'un outil à un autre ;

- Des concepts induits par le domaine d'application de l'environnement d'apprentissage. En effet, les EIAH pouvant être utilisés pour enseigner différentes disciplines pour différents niveaux. Par exemple, AMBRE-ADD (Duclosson, 2004) est utilisé dans l'enseignement des problèmes additifs à des élèves en primaire, Aplusix (Nicaud et al., 2002) est utilisé dans l'enseignement de l'algèbre à des collégiens et lycéens, Teleos (Luengo et al., 2006) est utilisé pour entraîner des

étudiants en médecine orthopédique. Une partie des concepts collectés dans les traces peut donc être spécifique au domaine d'application ;

- Des concepts induits par l'expérimentation d'apprentissage faisant usage d'un EIAH. En effet, suivant les objectifs de recherche, un même EIAH peut être utilisé dans des conditions différentes, par exemple dans un contexte d'apprentissage individuel ou collaboratif, en présentiel ou à distance ;

- Des concepts induits par l'analyse. Comme, nous l'avons déjà signalé, il est souvent utile de travailler de manière itérative (Dyke, 2009) et de capitaliser sur des analyses antérieures. Le chercheur peut donc avoir besoin d'interroger les ressources d'une analyse antérieure durant son travail. Par exemple, les concepts relatifs au diagnostic de l'activité d'apprentissage sont induits par l'analyse.

Cette diversité de l'origine des concepts explique le fait que l'ontologie reste ouverte, et peut être enrichie au fur et à mesure des besoins de représentation et d'analyse des chercheurs. Nous avons choisi de ranger les concepts du modèle sémantique de manière thématique sous des catégories.

5.4 Relations entre concepts

Dans la définition de la taxonomie des concepts, nous nous sommes fixés un objectif visant à éviter de proposer une structure compliquée et profonde des concepts. En effet, la définition d'une ontologie compliquée risque de nous renvoyer d'une certaine manière au problème d'une représentation partagée des données pouvant répondre à certains besoins et ne pas être adaptée à d'autres. L'ontologie proposée doit, au contraire, être simple et définir un nombre minimal de relations entre les concepts pour que son évolution et son enrichissement soient faciles à réaliser.

Nous définissons deux types de relations pouvant exister entre deux concepts de la taxonomie : la relation hiérarchique taxonomique classique et la relation d'agrégation.

5.4.1 Relation « est un » (*SubClassOf*)

Le premier type de relation est « est un » permettant d'exprimer la relation de subsomption entre concepts. Cette relation est très utilisée dans la définition de taxonomies

permettant de décrire une hiérarchie de concepts. Dans notre cas, il peut s'avérer utile de définir un concept comme une spécialisation (ou sous-concept) d'un autre concept pour préciser davantage la sémantique associée au concept. La primitive « rdfs:subClassOf » définie par le langage RDFS (RDFS, 2004) est utilisée pour représenter ce type de relation. Par exemple les concepts « expéditeur » et « destinataire » peuvent être définis comme des sous-classes du concept « participant ».

5.4.2 Relation « agrège » (*HasPart*)

Le deuxième type de relation est « agrège », et permet d'exprimer qu'un concept C_1 agrège un autre concept C_2 représentant des données utiles pour la description de C_1. Cela veut dire que les données décrites par C_2 définissent une partie de la sémantique des données décrites par C_1. Par exemple, le concept « interaction de chat » peut être défini comme l'agrégation, entre autres, des concepts « indicateur temporel », « expéditeur du message », et « message chat », qui représentent des données éventuellement présentes dans le traçage d'une interaction de chat. Il est à noter qu'un concept constituant peut lui-même être une agrégation d'autres concepts. Par exemple, le concept « message chat », constituant du concept « interaction de chat », peut lui-même rassembler les concepts « salon de clavardage », « identifiant du message », et « contenu du message ». L'utilisation de cette relation présente deux avantages grâce à sa généricité sémantique. Le premier avantage concerne les agrégations multiples entre concepts. En effet, un même concept peut faire partie de la définition de différents concepts. Par exemple, le concept « indicateur temporel » est un constituant des concepts « interaction de chat » et « interaction de forum ». Le deuxième avantage est lié à l'évolution de la taxonomie des concepts. La définition des concepts étant incrémentale et participative, l'ajout de nouveaux concepts et leur agrégation au sein d'autres concepts doit être facile à réaliser. Ainsi si un chercheur identifie un concept qu'il considère pertinent à intégrer à l'ontologie, il suffit de créer ce concept, de lui associer une définition sémantique et d'exprimer son agrégation éventuelle à des concepts existants de l'ontologie. Par ailleurs, un nouveau concept peut lui-même comporter des concepts précédemment définis dans l'ontologie. Le concept d'agrégation que nous considérons correspond à celui défini dans le paradigme objet. En effet, il exprime la définition d'un concept comme étant composé d'un ensemble d'autres concepts. Cependant, vu la grande variété des données tracées, l'agrégation peut être partiellement vérifiée d'un format de traces à un autre (p. ex. le concept « chat message » est composé des concepts « salon de clavardage », « id message » et

« contenu message » pour le format de traces F1, alors qu'il n'est composé que de « contenu message » pour le format F2). Notons que l'agrégation peut être liée à des contraintes. Un exemple de contrainte peut être l'exclusion, par exemple un concept ne peut pas être composé au même moment des concepts « indicateur de séquence » et « indicateur temporel ». D'autres types de contraintes pourraient être définis sous forme de règles plus complexes. Nous ne formalisons actuellement pas de contraintes dans la définition des agrégations de concepts. En effet, nous considérons l'agrégation comme une simple relation ensembliste définissant un concept comme l'agrégation d'un ensemble de concepts qui le composent.

5.5 Deux types de concepts

Nous avons identifié deux types de concepts pouvant être décrits dans la taxonomie : concepts simples et concepts complexes. Cette différenciation est liée à l'utilisation ou non de la relation « agrège » dans la définition du concept.

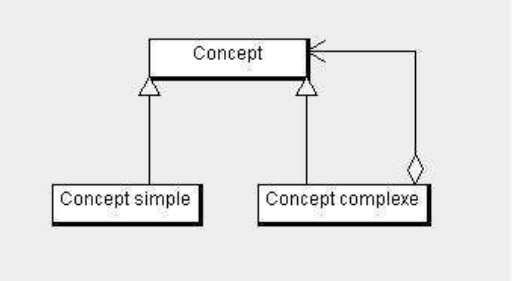


Figure 40 Types de concepts

5.5.1 Concept simple

Un concept est simple si sa définition ne contient aucune agrégation d'autres concepts (à l'aide de la relation « agrège »). Un concept simple désigne une donnée simple extraite à partir des ressources d'un corpus partagé. Un tel concept n'a pas besoin d'intégrer d'autres concepts pour spécifier plus en détail sa signification. Par exemple les concepts « id message » et « contenu message » sont des concepts simples car leur définition n'utilise pas d'agrégation d'autres concepts. Si un chercheur participant à l'évolution d'une ontologie exprime le besoin d'ajouter une agrégation dans la définition d'un concept simple, et que cette évolution est générique, l'ajout de l'agrégation à la définition du concept le rend complexe.

5.5.2 Concept complexe

Un concept complexe est défini comme l'agrégation d'autres concepts simples et/ou complexes. Il permet grâce à la relation « agrège » de définir un concept à partir d'autres concepts. Cela permet de composer à partir de concepts de niveaux sémantiques plus bas, d'autres de niveaux sémantiques plus élevés. Cette décomposition permet l'intégration d'un même concept à plus d'un concept de niveau sémantique plus élevé. Par exemple le concept « expéditeur » est un constituant des concepts « interaction de chat » et « interaction de forum ».

Un concept complexe défini dans l'ontologie peut être utilisé dans l'interrogation des ressources contenues dans les corpus partagés. Il est défini de la manière la plus générique possible pour permettre l'interrogation d'un nombre maximum de ressources de différents corpus contenant des données ayant des sémantiques proches. Les concepts constituant un concept complexe peuvent donc être partiellement retrouvés dans une ressource interrogée (par exemple les traces d'interaction de chat ne contiennent pas toutes une donnée relative au salon de clavardage où le chat a lieu). Pour cette raison, un concept complexe est défini comme l'agrégation de l'ensemble de tous les concepts pouvant le décrire. Cependant, l'instance d'un concept complexe utilisé dans un contexte particulier peut n'utiliser qu'un sous-ensemble des concepts qui le constituent.

5.6 Catégorisation des concepts

La définition d'une ontologie des concepts interrogeables des corpus de traces contextualisées et enrichies par des analyses est un processus complexe. Les EIAH pouvant traiter de domaines d'application très variés, les concepts définis dans l'ontologie sont définis d'une manière incrémentale au fur et à mesure de l'évolution des besoins d'interrogation des contenus des corpus partagés.

Les concepts relatifs aux données partagées dans un corpus et dont l'interrogation peut être intéressante pour l'analyse doivent être définis d'une manière thématique permettant de faciliter la navigation dans la liste des concepts définis. Comme première proposition, nous suggérons six catégories principales de concepts pouvant être retrouvés dans les corpus de traces d'interaction d'apprentissage. Ces catégories ne sont pas exhaustives mais donnent un

premier exemple de classification. Elles peuvent être enrichies au fur et à mesure de l'évolution des besoins.

5.6.1 Concepts relatifs à l'aspect situé des interactions tracées

Une trace est composée d'objets situés les uns par rapport aux autres. La première catégorie définit des concepts relatifs à l'aspect situé des interactions tracées et qui expriment une relation d'ordre entre ces interactions. La relation d'ordre peut être exprimée de deux manières : (1) à l'aide d'un estampillage temporel, ou (2) grâce au caractère séquentiel des interactions exprimé par une position séquentielle. La Figure 41 illustre les concepts que nous proposons pour cette catégorie. Nous définissons six concepts dont deux concepts complexes et quatre simples. Les concepts simples sont :

- « estampille temporelle de début » (BeginTimestamp) qui fait référence à un estampillage temporel marquant le début d'une interaction tracée ;
- « estampille temporelle de fin » (EndTimestamp) qui permet de marquer la fin d'une interaction ;
- « durée » (Duration) exprimant la durée associée à une interaction ;
- et « position séquentielle » (SequentialPosition) qui associe un rang à une interaction observée permettant de la situer par rapport à d'autres.

Les deux concepts complexes sont :

- « indicateur temporel » (TemporalIndicator) (cf. Figure 41a) qui est l'agrégation des concepts « estampille temporelle de début », « durée », et « estampille temporelle de fin » ;
- et « indicateur de séquence » (SequentialIndicator) (cf. Figure 41b) qui est l'agrégation des concepts « position séquentielle » et « durée ».

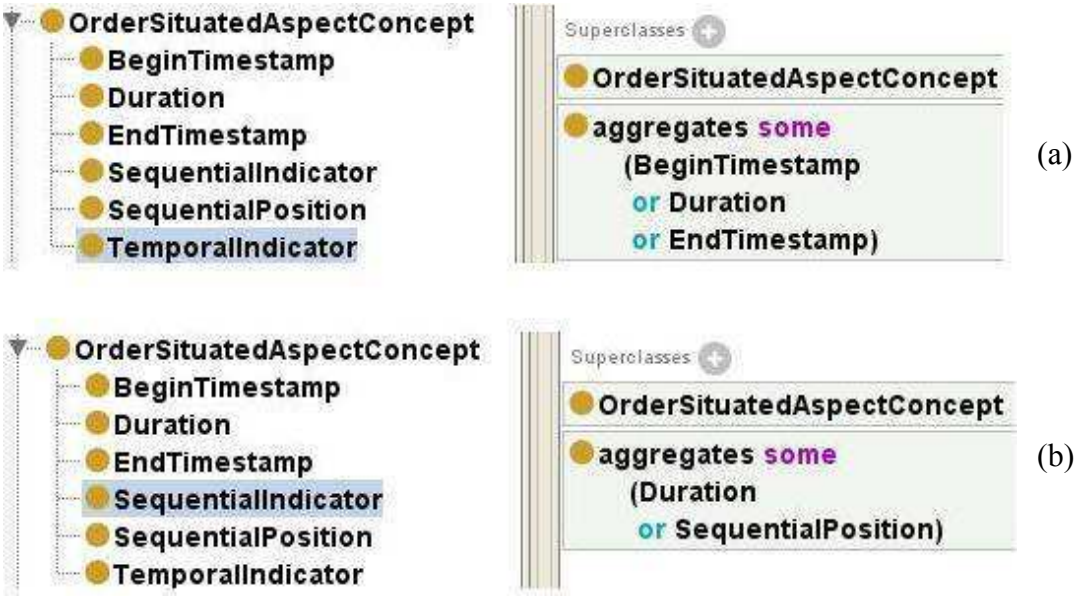


Figure 41 Concepts relatifs à l'aspect situé des interactions tracées

5.6.2 Concepts relatifs au contexte d'apprentissage

Un concept relatif au contexte d'apprentissage permet de décrire une donnée générale concernant une situation d'apprentissage. Nous avons défini cinq concepts simples relatifs au contexte d'apprentissage (cf. Figure 42) qui peuvent éventuellement devenir complexes par l'agrégation d'autres concepts définis ultérieurement. Ces concepts sont :

- « cours » (course), c'est un concept qui peut éventuellement être présent dans la trace pour dénoter le cours auquel appartient l'interaction tracée ;
- « problème » (Problem), ce concept désigne un problème en cours de résolution lors de l'interaction tracée ;
- « type d'événement » (EventType), ce concept identifie le type de l'événement tracé, par exemple dans un outil de type chat, une interaction peut correspondre à trois différents types d'événements qui sont la connexion, la déconnexion ou l'envoi d'un message ;
- « session », ce concept peut se retrouver dans la trace lorsque l'activité est organisée en sessions ;
- et « outil utilisé » (UsedTool), un EIAH offre parfois différents types d'outils permettant des activités différentes, un tel concept peut se retrouver dans une interaction tracée pour référencer l'outil utilisé lors de sa génération.

Ces concepts peuvent être enrichis pour identifier des concepts contextuels retrouvés dans les corpus et utiles pour l'interrogation.

Figure 42 Concepts relatif au contexte d'apprentissage

5.6.3 Concepts relatifs au participant

La troisième catégorie concerne les concepts relatifs aux participants à l'interaction. De telles données sont souvent utilisées pour étudier l'activité d'un participant particulier ou d'un ensemble de participants. Nous définissons donc le concept « participant » (cf. Figure 43) comme l'agrégation des concepts :

- « identifiant du participant » (ParticipantID), qui permet d'associer un identifiant unique à un participant ;
- « nom d'utilisateur du participant » (ParticipantUsername), est un nom choisi par le participant pour s'identifier, l'identifiant et le nom d'utilisateur sont censés être anonymes et donc ne permettent pas de divulguer l'identité du participant ;
- « groupe du participant » (ParticipantGroup), ce concept est utilisé dans le cadre d'une activité collective et permet d'identifier le groupe auquel appartient le participant ;
- « rôle du participant » (ParticipantRole), ce concept est aussi utilisé dans le cadre d'une activité collective et exprime le rôle attribué à un participant dans une activité d'apprentissage ;
- et « classe du participant » (ParticipantLevel), ce concept exprime le niveau d'étude auquel appartient le participant.

Le concept « participant » est un concept générique qui référence un acteur participant à une situation d'apprentissage dont l'activité est tracée. Dans une situation d'activité de communication, un participant peut être l'« expéditeur » ou le « destinataire » dans une interaction. Nous définissons donc les concepts « expéditeur » et « destinataire » comme des sous-classes de participant (à l'aide de la relation de spécialisation « est un »). Un « destinataire » est spécialisé à son tour par un « destinataire principal », un « destinataire en

copie », et un « destinataire en copie cachée ». Les concepts constituant le concept « participant » le sont aussi pour les cinq concepts qui spécialisent ce concept (grâce au principe de l'héritage). Les concepts « expéditeur » et « destinataire », ainsi que ceux qui spécialisent ce dernier, se trouvent classés sous cette catégorie de par la relation de spécialisation qui les lie avec le concept « participant », mais thématiquement ils sont plutôt classés sous la catégorie des concepts relatifs à la communication présentée dans le paragraphe suivant.

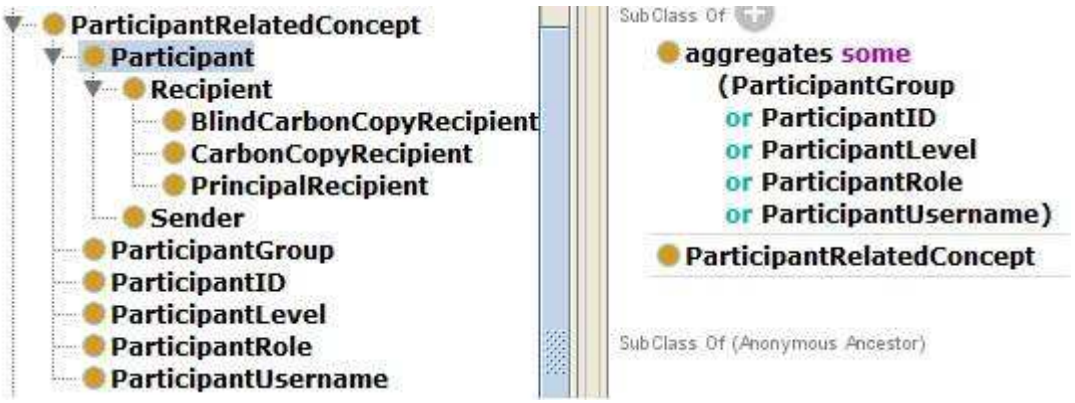


Figure 43 Concepts relatifs au participant

5.6.4 Concepts relatifs à la communication

Cette catégorie concerne les concepts relatifs aux données retrouvées dans les traces d'interaction relatives à l'utilisation d'outils de communication (cf. Figure 44) tels qu'un chat et un forum. Nous définissons différents concepts simples et complexes relatifs aux traces de communication. Par exemple le concept complexe « interaction de chat » (ChatInteraction) peut être défini comme l'agrégation des concepts :

- « message de chat » (ChatMessage), concept complexe (cf. Figure 45) qui est composé à son tour des concepts :
  - « identifiant du message » (MessageID) permettant d'identifier d'une manière unique le message en question,
  - « salon de clavardage » (Chatroom) où le chat a lieu,
  - et « contenu du message » (MessageContent), le texte envoyé dans le chat ;
- « cours » (Course), au cours duquel l'activité de communication en utilisant le chat est programmée, ce concept est défini parmi les concepts relatifs au contexte d'apprentissage (cf. paragraphe 5.6.2), les quatre concepts suivants le sont aussi ;

- « problème » (Problem), le problème en cours de résolution auquel la communication est liée ;
- « type de l'événement » (EventType), indique le type d'événement de l'interaction tracée ;
- « session », indique la session au cours de laquelle l'interaction est réalisée ;
- « outil utilisé » (UsedTool), indique l'outil utilisé durant l'interaction ;
- « expéditeur » (Sender), le participant (concept défini dans la catégorie des concepts relatifs à un participant) ayant envoyé le message, ce concept appartient à cette catégorie des concepts relatifs à la communication ;
- et les concepts « indicateur temporel » (TemporalIndicator) et « indicateur de séquence » (SequentialIndicator) permettant d'exprimer le caractère situé d'une interaction présentés dans le paragraphe 5.6.1.

Il est donc à noter qu'un concept appartenant à une certaine catégorie peut être défini comme l'agrégation de concepts classés sous d'autres catégories. Comme l'exemple du concept « interaction de chat » (ChatInteraction) qui est composé de trois concepts relatifs au contexte d'apprentissage et deux concepts relatifs au caractère situé de l'interaction.

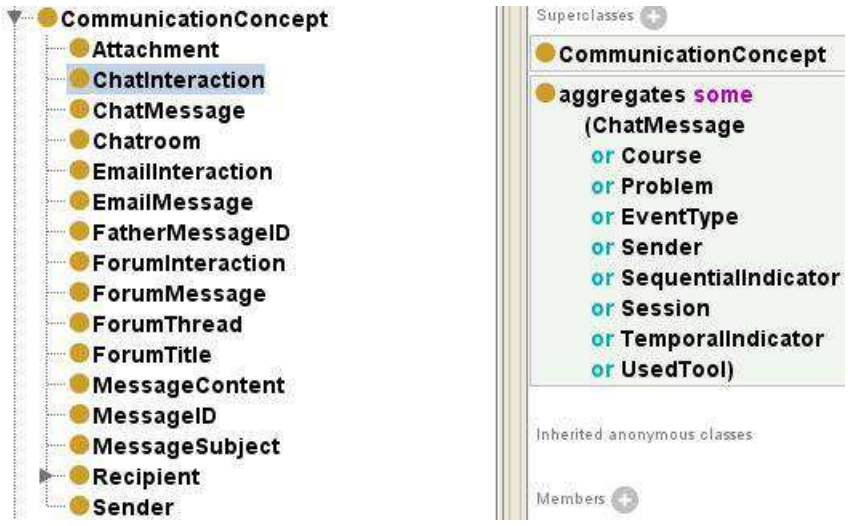


Figure 44 Concepts relatifs à la communication, et exemple du concept complexe « interaction de chat »

Figure 45 Le concept « message de chat » et les concepts qui le constituent

5.6.5 Concepts relatifs à la production

Cette catégorie concerne les concepts retrouvés dans les traces d'interaction avec un outil de production offert à un participant pour son activité d'apprentissage. Un outil de production peut-être destiné à une activité de production individuelle (par exemple un tuteur intelligent) ou collective (éditeur de texte partagé). Le concept complexe « interaction de production » (cf. Figure 46) est composé d'un ensemble de concepts dont certains font partie de cette catégorie de concepts relatifs à la production, et d'autres classés sous les autres catégories. Le concept « interaction de production » (ProductionInteraction) peut être défini comme l'agrégation des concepts :

- « réponse » (Answer), ce concept désigne une réponse donnée par un participant lors de son activité d'apprentissage,
- « objet produit » (ProducedObject), les outils de production permettent souvent, et suivant le domaine d'application de l'EIAH de produire des objets liés à l'activité d'apprentissage (par exemple des composants électriques dans le micro-monde TPElec[10] pour l'apprentissage de l'électricité). Ce concept (cf. Figure 47) est composé des concepts :
  - « identifiant de l'objet produit » (ProducedObjectID),
  - « nom de l'objet produit » (ProducedObjectName),
  - « type de l'objet produit » (ProducedObjectType), ce concept peut être lié au domaine d'application de l'EIAH,
  - et « contenu de l'objet produit » (ProducedObjectContent).

Ces deux concepts (« réponse » et « objet produit ») sont classés sous la présente catégorie des concepts relatifs à la production ; Le concept « interaction de production » est composé également des concepts suivants :

Cinq concepts de la catégorie relative au contexte d'apprentissage :

- « cours » (Course),
- « problème » (Problem),
- « type d'événement » (EventType),

[10] http://tpelec.imag.fr

– « session »,

– et « outil utilisé » (UsedTool).

– le concept « participant », de la catégorie des concepts relatifs aux participants,

Deux concepts de la catégorie des concepts relatifs à l'aspect situé des interactions :

– « indicateur temporel » (TemporalIndicator),

– et « indicateur de séquence » (SequentialIndicator).

Et trois concepts de la catégorie des concepts liés au diagnostic des connaissances (présentée dans le paragraphe suivant) :

– « connaissance » (Knowledge),

– « évaluation » (Evaluation),

– et « conseil » (Advice).

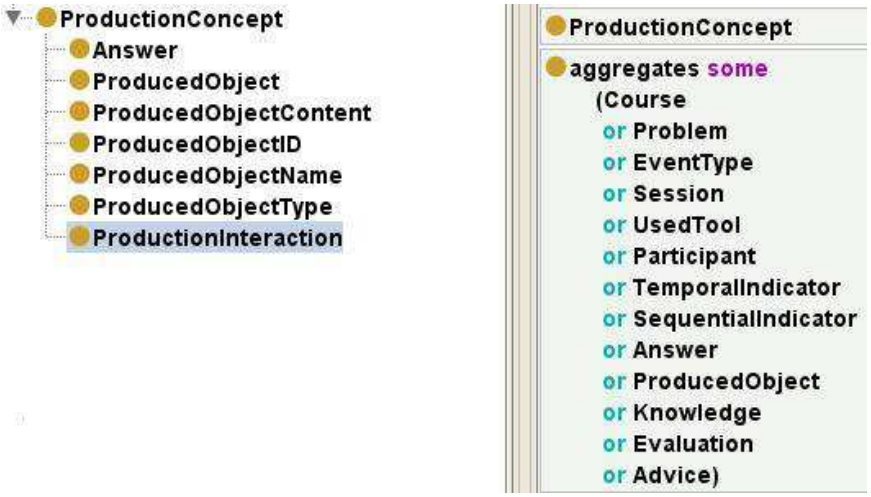


Figure 46 Concepts relatifs à la production

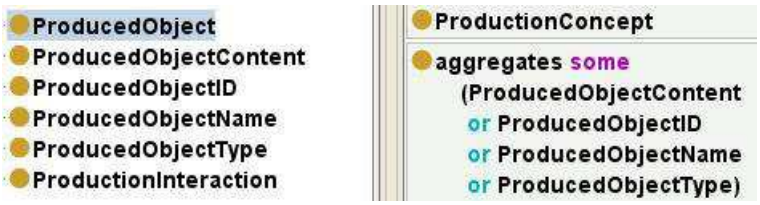


Figure 47 Le concept « objet produit » et les concepts qui le constituent (à droite)

5.6.6 Concepts relatifs au diagnostic

Certains EIAH utilisent les traces d'interaction et les collectent en temps réel pour diagnostiquer l'activité d'apprentissage, ce qui permet de savoir si oui ou non un participant a

acquis une connaissance, et de le conseiller éventuellement pour l'aider dans son activité. Ce genre de fonctionnalité est généralement offert par les tuteurs intelligents. Les traces collectées à la fin d'une situation d'apprentissage peuvent donc contenir des données relatives au diagnostic. De telles données peuvent servir dans le cadre d'analyses différées de traces, par exemple pour comparer le diagnostic automatique à celui fait par un humain, ou pour évaluer différentes techniques de diagnostic (Lalle et al., 2013). Nous avons identifié trois concepts :

- « connaissance » (Knowledge), correspond à la connaissance mise en jeu dans l'activité de l'apprenant et dont l'acquisition est étudiée ;
- « évaluation » (Evaluation), correspond à l'évaluation calculée suite au diagnostic de l'activité, ce concept peut être calculé par l'outil d'apprentissage lui-même, par un expert humain, ou par des outils de diagnostic externes. Il permet d'enregistrer le résultat de l'évaluation de l'activité d'un apprenant. Un exemple d'évaluation peut être « correct » ou « incorrect » ou un pourcentage relatif au degré d'acquisition de la connaissance en jeu ;
- et « conseil » (Advice), correspond à un conseil formulé par le système comme un feedback permettant de guider l'apprenant dans son activité.

Figure 48 Concepts relatifs au diagnostic et au feedback

5.7 Synthèse

Ce deuxième modèle sémantique des concepts interrogeables d'un corpus partagé représente un moyen pour les chercheurs d'avoir une sémantique partagée concernant les concepts interrogeables des corpus. C'est un modèle ouvert qui peut être facilement enrichi en ajoutant de nouveaux concepts et en les liant aux concepts existants à l'aide des relations de subsomption et d'agrégation. Ce modèle peut être vu comme un moyen de communication sémantique permettant aux chercheurs d'aligner les concepts relatifs aux contenus de leurs corpus avec ceux définis dans l'ontologie. Il permet également d'interroger les corpus

existants en se basant sur un vocabulaire partagé. Enfin ce modèle permet d'établir une interopérabilité sémantique entre les corpus et les outils d'analyse. Le modèle sémantique étant défini d'une manière incrémentale et participative, il peut évoluer en fonction des besoins d'analyse. Cette évolution s'appuie sur la réflexion présentée dans le paragraphe 5.3 traitant de la genèse des concepts du modèle et l'extension de l'ontologie décrite dans le chapitre 7 permettant d'envisager une progression harmonieuse. Le modèle sémantique présenté dans ce chapitre représente pour nous l'alternative que nous proposons pour éviter la proposition d'une nouvelle représentation pour la structuration des traces collectées. En construisant la partie de l'ontologie relative à ce modèle sémantique, nous nous sommes basés sur les travaux existants pour identifier les données récurrentes que l'on retrouve généralement dans les traces collectées. Par exemple, pour identifier les concepts relatifs à la communication, nous nous sommes inspiré des formats proposés par le projet MULCE (Reffay et al., 2008), TraVis (May et al., 2008), et CALICO (Giguet et al., 2009). Il est important de rappeler que ce modèle a encore besoin d'être enrichi pour couvrir le maximum de concepts nécessaires pour les analyses en EIAH. Ce modèle sémantique contribue à la flexibilité de l'approche en permettant le partage de corpus hétérogènes sans imposer de contraintes sur les formats de représentation des données.

Chapitre 6 : L'approche « Proxyma » : un modèle opérationnel pour l'interrogation des corpus

6.1 Introduction

Dans ce chapitre, nous présentons le modèle opérationnel. Ce troisième modèle de l'approche « Proxyma » permet l'interrogation de la base des corpus partagés. Il définit un ensemble d'opérations permettant l'extraction, la conversion, le filtrage et le formatage des données des corpus afin de les analyser. En se basant sur le modèle sémantique, les scripts relatifs aux opérations du modèle opérationnel permettent de réaliser l'un des objectifs fondamentaux de l'approche « Proxyma » relatif à l'interfaçage des corpus et des outils d'analyse partagés. Ces scripts permettent de faire correspondre une partie des données d'un corpus avec le couple sémantique/représentation (S_o,R_o) relatif à un outil d'analyse, en se basant sur les concepts définis par le modèle sémantique. Comme son nom l'indique, le modèle opérationnel incarne l'aspect opérationnel de l'approche « Proxyma ». Il définit six types d'opérations relatifs aux mécanismes opérationnels pour l'exploitation des contenus des corpus. Ces différentes opérations sont présentées dans la suite.

6.2 Interrogation de concept

L'interrogation de concept joue un rôle d'alignement entre un concept interrogeable défini dans l'ontologie, et la donnée qui lui correspond dans une ressource partagée dans un corpus. Il s'agit, à partir d'un concept défini dans l'ontologie, de chercher dans un corpus particulier les incarnations de ce concept dans leur représentation propre au corpus. Une telle opération permet d'interroger trois types de ressources :

- les ressources de type traces structurées pouvant être traitées automatiquement,
- les ressources traces enrichies par des analyses antérieures (ressources produites par l'analyse),
- et des ressources contenant des données contextuelles utiles pour l'analyse (ressources utilisées durant l'analyse).

Un script d'interrogation de concept interroge un concept simple ou un concept complexe (cf. Figure 49). Dans le cas d'interrogation d'un concept complexe C, le script correspondant fait appel aux scripts d'interrogation des concepts qui constituent C. Un script d'interrogation d'un concept simple interroge un concept simple défini dans l'ontologie et est lié à une structure particulière définissant l'organisation des données dans la ressource interrogée (cette structure peut être plus ou moins formelle, par exemple sous forme d'un schéma XSD définissant le schéma d'un document XML, ou exprimée sous une forme moins formelle (par exemple textuelle) définissant la sémantique des données de la ressource interrogée).

Notons que l'interrogation d'un concept peut nécessiter des conditions d'extraction suivant la répartition des données interrogées. Par exemple, si une ressource traces contient la donnée « identifiant du participant », et qu'une ressource contextuelle contient en plus la donnée « rôle du participant » et que cette deuxième donnée est utile pour l'analyse, alors l'opération d'interrogation du concept « rôle du participant » doit recevoir en entrée la donnée correspondant au concept « identifiant du participant » qui permettra de faire la jointure et d'extraire la donnée attendue. L'interrogation de concept est exécutée sur une ressource contenant les données interrogées et renvoie tous les enregistrements correspondant au concept interrogé. La Figure 50 et la Figure 51 ci-dessous illustrent respectivement un

exemple de script d'interrogation du concept simple « rôle du participant », et un exemple de script d'interrogation du concept complexe « expéditeur » (ce dernier étant un sous-concept de « participant »). Ces scripts sont liés aux traces de forum générées par la plateforme d'apprentissage Moodle. Cette dernière ainsi que d'autres exemples de scripts seront présentés dans le chapitre 9 traitant des exemples d'application de l'approche « Proxyma ».

Ce type d'opération permet donc d'aligner les concepts définis par l'ontologie et partagés entre les chercheurs avec les données contenues dans les ressources interrogeables des corpus (dont les contenus sont définis par des structures particulières). L'avantage de cette opération est sa flexibilité par rapport aux types de concepts que nous avons définis (simple et complexe) en permettant qu'une opération interrogeant un concept complexe appelle une autre interrogeant un concept constituant le concept complexe.

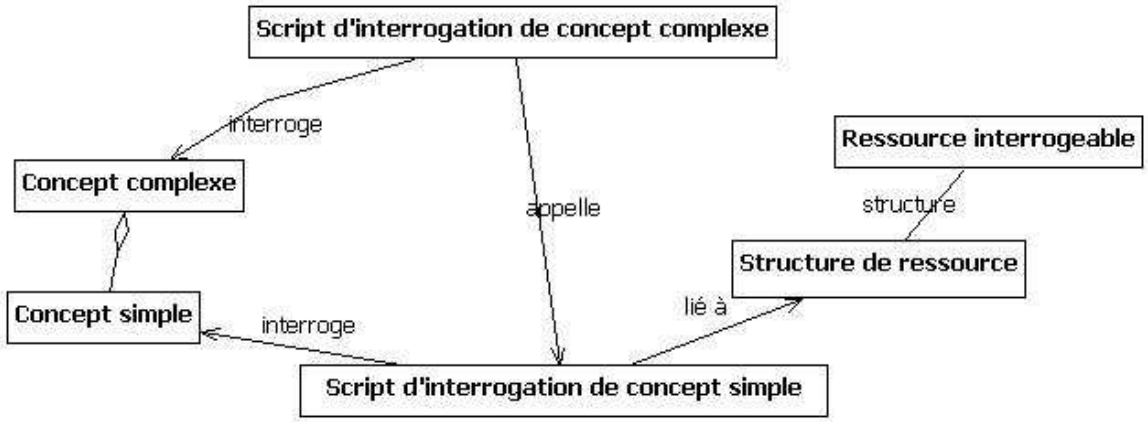


Figure 49 Script relatif à l'opération d'interrogation de concept

Simple concept querying script of Moodle forum traces

```
declare function moodle:forumMessageSenderRole($sendersIDs as xs:string*, $usersFile as
xs:string, $rolesFile as xs:string)
{
   for $i in $sendersIDs
   let $roleID:=doc($usersFile)//table/column[@name="userId"]/text()[. = $i]
         /../../column[@name="userRoleId"]/text()
   let $roleName := doc($rolesFile)//table/column[@name="roleId"]/text()[.=$roleID]/../../
         column[@name="roleName"]/text()
   return $roleName
};
```

Figure 50 Exemple de script d'extraction du concept simple « rôle de l'expéditeur » à partir des traces de forum de Moodle

Complex concept querying script of Moodle forum traces

```
declare function moodle:forumMessageSender($postsFile as xs:string, $usersFile as xs:string, $rolesFile as xs:string)
{
  let $messagesSendersIds := moodle:forumMessageSenderID($postsFile)
  let $messagesSendersRoles := moodle:forumMessageSenderRole($messagesSendersIds,$usersFile,$rolesFile)
  for $i at $j in $messagesSendersIds
  return
  <Sender>
  {
      <ParticipantID>{$messagesSendersIds[$j]}</ParticipantID>,
      <ParticipantRole>{$messagesSendersRoles[$j]}</ParticipantRole>
  }
  </Sender>
};
```

Figure 51 Exemple de script d'extraction du concept complexe « expéditeur » à partir des traces de forum de Moodle

6.3 Conversion de type de données

Une donnée extraite par un script d'interrogation d'un concept simple à partir des ressources interrogeables des corpus peut être exprimée dans un type de données différent de celui attendu par un outil d'analyse. Une conversion est alors nécessaire pour exprimer la donnée extraite dans le type de donnée adéquat. Un script de conversion de type de données (cf. Figure 52) est défini pour un couple : type de données en entrée et type de données en sortie. Un exemple typique concerne le type de données utilisé dans la représentation de la date. En effet, certains systèmes représentent la date sous forme de temps Unix exprimée par une valeur entière, alors que d'autres la représentent avec le type date/heure. Un script de conversion peut donc prendre en entrée une date exprimée en temps Unix et la convertir en date/heure et inversement (cf. Figure 53).

Un script de conversion de type de données peut être appelé au besoin par un script d'extraction. L'opération de type extraction est présentée dans le paragraphe suivant.

Figure 52 Script relatif à l'opération de conversion de type de données

Datatype converting script (unix timestamp to date-time)

```
declare function utils:unix-to-dateTime($v) as xs:dateTime
{
  xs:dateTime("1970-01-01T00:00:00-00:00")
  + xs:dayTimeDuration(concat("PT", $v, "S"))
};
```

Figure 53 Exemple de script de conversion de type de donnée, ce script prend un timestamp unix en entrée et le convertit au format dateTime

6.4 Extraction

Un script d'extraction permet, à partir des ressources interrogeables d'un corpus, d'extraire les données équivalentes à un concept complexe défini dans l'ontologie et utile pour l'analyse avec un outil partagé. Le script d'extraction (cf. Figure 54) appelle le script d'interrogation relatif au concept complexe à interroger. Le script d'extraction est donc, comme l'est le script d'interrogation de concept, lié à une (ou plusieurs) structure définissant l'organisation des données dans la ou les ressources interrogées. Un script d'extraction permet d'interroger un concept complexe pour extraire les données attendues en entrée par un outil d'analyse. Le script d'extraction est donc lié à une structuration des données d'entrée de l'outil d'analyse. Cette structure peut définir des types de données que doivent avoir les données extraites. Le script d'extraction peut donc faire appel à des scripts de conversion de types de données permettant de convertir les données extraites dans les types de données attendues, par exemple, extraire les données relatives au concept complexe « interaction de chat » en convertissant la donnée relative au concept « estampille temporelle de début » qui est de type temps Unix au type date/heure.

La Figure 55 ci-dessous illustre un exemple de script d'extraction relatif au concept complexe « interaction dans le chat » à partir des traces de l'environnement d'apprentissage DREW. Ce dernier ainsi que d'autres exemples de scripts seront présentés dans le chapitre 9.

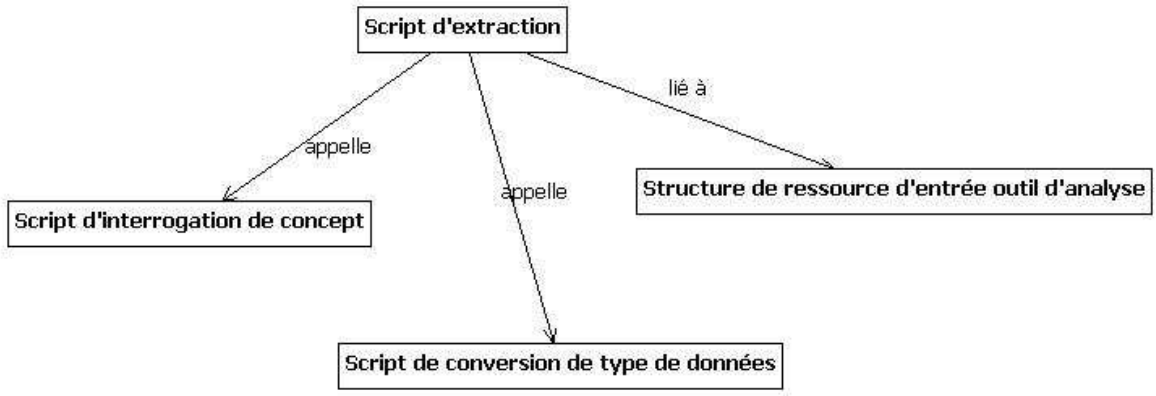


Figure 54 Script relatif à l'opération d'extraction

Extracting script – DREW chat messages

```
declare function drew:chatInteractions($docs as xs:string*) as node()*
{
    for $i in $docs
    return
    drew:chatInteraction($i)
};

declare function drew:chatInteraction($doc as xs:string) as node()*
{
    let $chatMessagesSenders := drew:chatMessageSender($doc), $chatTemporalIndicator :=
          drew:chatTemporalIndicator($doc),
    $chatMessages := drew:chatMessage($doc)
    for $i at $c in $chatMessages
    return
    <ChatInteraction>
    {
       $chatMessagesSenders[$c],
       $chatTemporalIndicator[$c],
       $chatMessages[$c]
    }
    </ChatInteraction>
};
```

Figure 55 Exemple de script d'extraction relatif au concept complexe « interaction dans le chat » à partir des traces de l'environnement DREW

6.5 Filtrage

Les données extraites à l'aide d'un script d'extraction contiennent toutes les données relatives aux concepts constituant le concept complexe interrogé. Il peut s'avérer utile de filtrer les données extraites par un script d'extraction pour s'adapter à des besoins d'analyse particuliers. Un script de filtrage (cf. Figure 56) s'exécute donc sur le résultat d'un script d'extraction. Il est aussi possible d'exécuter un script de filtrage sur le résultat d'un autre script de filtrage pour ne garder qu'une partie des résultats. Un script de filtrage permet de

définir deux types de filtres. Le premier type de filtre permet de choisir les concepts de projection parmi les concepts simples et complexes qui constituent le concept complexe interrogé par le script d'extraction. Ce filtre permet de restreindre les données qui seront présentées à l'entrée de l'outil d'analyse. Ce filtre correspond à la clause « Select » du langage SQL d'interrogation de bases de données relationnelles. Par exemple, un script d'extraction sur le concept « interaction de chat » (cf. Figure 44), renvoie les différentes données relatives aux concepts qui le constituent : « message de chat », « expéditeur », et « indicateur temporel ». Le script de filtrage renvoie les données relatives aux concepts « contenu du message » (constituant du concept « message de chat »), « nom d'utilisateur » (constituant du concept « expéditeur »), et « estampille temporelle de début » (constituant du concept « indicateur temporel »).

Le deuxième type de filtre permet de définir des conditions de sélection sur les données retournées par le script d'extraction. Une condition de sélection est définie sur un concept simple constituant le concept complexe interrogé par le script d'extraction, et permet de restreindre les enregistrements (nombre de lignes) à retourner. Une telle condition correspond aux restrictions exprimées dans une clause « Where » du langage SQL. Une condition de sélection est donc définie par un triplet (concept simple, opérateur, valeur de sélection). L'opérateur permet la comparaison (par exemple : « égal à », « différent de », « supérieur à ») entre la valeur retrouvée dans les données extraites et la valeur de sélection définie par la condition de sélection. Un exemple de condition de sélection peut être associé au concept « estampille temporelle de début » qui doit être comparé à une date particulière par l'opérateur « supérieur ou égal ».

Notons que les deux types de filtres peuvent être utilisés d'une manière indépendante. En effet, il est possible de définir un filtre sur les concepts de projection et ne pas définir de condition de sélection. Si une ou plusieurs conditions de sélection sont définies et qu'aucune définition de concept de projection n'est explicitée, cela veut dire que le résultat du script de filtrage contiendra, dans le résultat de son exécution, tous les concepts constituant le concept complexe interrogé par le filtre d'extraction. Par ailleurs, les conditions de sélection peuvent être définies pour des concepts simples qui font ou non partie des attributs de projection.

La Figure 57 ci-dessous illustre un exemple de script de filtrage permettant de retourner les interactions de chat réalisées par un utilisateur particulier.

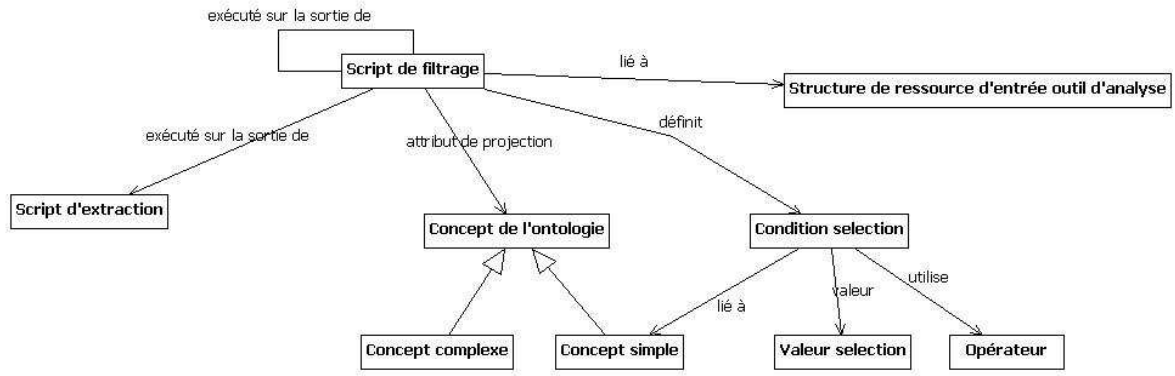


Figure 56 Script relatif à l'opération de filtrage

Filtering script – DREW chat message sent by a user

```
declare function drew:chatInteractionByUser($chatInteractions as node()*, $user as xs:string) as node()*
{
    for $i at $c in $chatInteractions
    return
    if($i/Sender/ParticipantUserName/text()[.=$user])
    then
    <ChatInteraction>
    {
       $chatMessagesSenders[$c],
       $chatTemporalIndicator[$c],
       $chatMessages[$c]
    }
    </ChatInteraction>
    else
    ()
};
```

Figure 57 Exemple de script de filtrage permettant de retourner les interactions de chat réalisées par un utilisateur particulier

6.6 Formatage

L'objectif étant d'assurer une interopérabilité entre les données partagées dans les corpus et les outils d'analyse, les données extraites et filtrées doivent être formatées pour être exploitables par un outil d'analyse. Un script de formatage (cf. Figure 58) est lié à la structure des données attendues par l'outil d'analyse. Il est exécuté sur la sortie d'un script de filtrage ou directement sur la sortie d'un script d'extraction si les données extraites n'ont pas besoin d'être filtrées. Un script de formatage peut aussi définir des conditions de tri des données formatées. De telles conditions peuvent être définies sur des concepts simples. Par exemple, trier les données relatives aux interactions de chat par rapport au concept « nom d'utilisateur », ce qui permet de regrouper les interactions par rapport à leur expéditeur.

La Figure 59 ci-dessous illustre un exemple de script de formatage réalisant la mise en forme de données relatives à des interactions de chat extraites à partir d'un corpus pour les analyser en utilisant l'outil Tatiana.

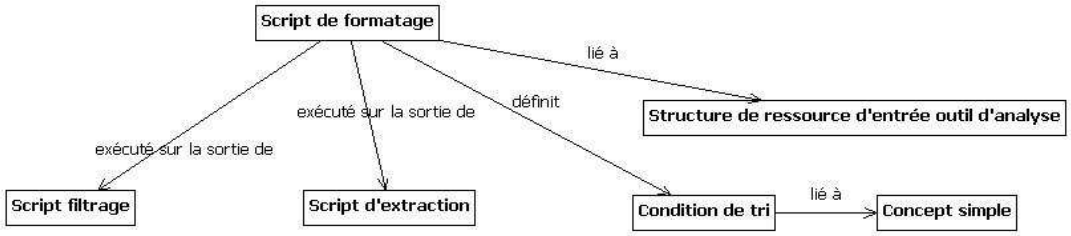


Figure 58 Script relatif à l'opération de formatage

Formatting script – chat interactions for Tatiana input

```
import module namespace
    jj = "http://kumquat.emse.fr/utilitaires"
    at "jjutils.xq" ;
          import module namespace
    util = "http://www.example.org/AnalysisTools/TatianaAnalysisTool"
    at « utils.xq" ;

    <display>
    {
        let $t := $arguments[1]
        let $d := doc($t)//ChatInteraction

        for $i at $j in $d
        return
        <item>
            <info name="src-anchor">
               <anchor>{
                    <doc>{ $t }</doc>,
                    <path>{jj:build-Path($i)}</path>
               }</anchor>
            </info>
            <info name="time">
                <time>
                   <date>{$i/TemporalIndicator/BeginTimestamp/text()}</date>
                   {if(not(empty($i/TemporalIndicator/Duration/text())))
                    then
                      <duration>{$i/TemporalIndicator/Duration/text()}</duration>
                    else ()}
                </time>
            </info>
           {
           util:recursiveRetrieving($i)
           }
        </item>
    }
    </display>
```

Figure 59 Exemple de script de formatage réalisant la mise en forme de données relatives à des interactions de chat pour les analyser en utilisant l'outil Tatiana

6.7 Fusion

Suivant les objectifs de l'analyse, et les données mises à disposition dans les corpus partagés, un chercheur peut être intéressé par l'analyse, en même temps, de données

provenant de corpus différents. Pour permettre au chercheur de travailler sur des données provenant de sources différentes et de les analyser sous forme d'un même flux de données, nous proposons le sixième type d'opération : la fusion. Il est évident que pour fusionner des données provenant de différentes sources, celles-ci doivent correspondre au même concept complexe interrogé par le script d'extraction. Les données résultantes des scripts d'extraction ou de filtrage (si un filtrage des données est nécessaire) appliqués aux données doivent être homogènes pour que leur formatage à l'aide du script de formatage donne des ressources ayant la structure attendue par l'outil d'analyse utilisé. Le rôle du script de fusion (cf. Figure 60) consiste donc à fusionner les données résultantes de l'exécution d'un script de formatage sur des données provenant de sources différentes et à les trier en définissant des conditions de tri.

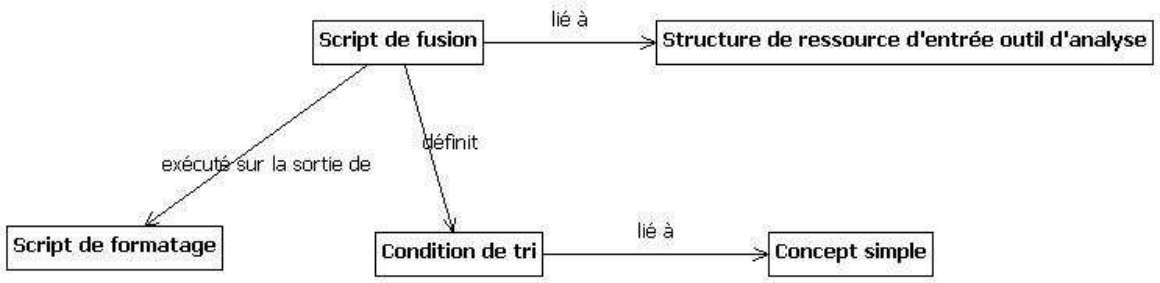


Figure 60 Script relatif à l'opération de fusion

6.8 Exemple de requête

Nous présentons un exemple assez complet permettant d'illustrer l'utilisation des différents types d'opérations. Nous considérons la requête suivante à exécuter sur le corpus « EMSE-LEAD » (présenté dans la section 9.4) : « retourner les interactions de chat dans lesquelles le message, envoyé par les étudiants « François » et « Aurélia », a une taille de trente caractères ou plus et contient le mot 'program' ». Le concept complexe qu'on interroge dans ce cas est « interaction de chat ». Ce dernier est composé des concepts « expéditeur », « indicateur temporel », et « message de chat ». La première étape consiste à écrire les scripts d'interrogation (cf. Figure 61) des concepts permettant de faire l'alignement entre les concepts définis dans le modèle sémantique et les données contenues dans les ressources du corpus. La deuxième étape consiste à définir le script d'extraction (cf. Figure 62), celui-ci fait appel au script d'interrogation du concept complexe « interaction de chat », et fait appel au

script de conversion de la date unix en type « dateTime ». La troisième étape consiste à définir un script de filtrage permettant d’extraire les messages ayant une taille supérieur ou égale à un nombre N de caractères. Ensuite un autre script de filtrage exécuté sur le résultat du précédent permet de ne garder que les messages envoyés par un utilisateur particulier. Un autre script de filtrage est également défini pour filtrer les résultats du précédent et garder uniquement les messages contenant une chaîne de caractères donnée. Ces différents scripts de filtrage sont illustrés sur la Figure 63. Enfin, un script de formatage doit être défini pour permettre de préparer les données extraites à l’entrée de l’outil d’analyse Tatiana (script illustré sur la Figure 59). Ces différents scripts sont génériques et réutilisables. Ces scripts étant probablement déjà définis au moment de l’interrogation d’un corpus, un chercheur souhaitant exécuter une requête peut n’avoir qu’à réutiliser des scripts déjà définis en fournissant les valeurs de paramètres nécessaires.

Script d'interrogation du concept simple « BeginTimestamp »

```
declare function drew:chatTemporalIndicatorBeginTimestamp($doc as xs:string) as xs:string*
{
    (doc($doc)//chat)/../time/date/text()
};
```

Script d'interrogation du concept simple « Duration »

```
declare function drew:chatTemporalIndicatorDuration($doc as xs:string) as xs:string*
{
    (doc($doc)//chat)/../time/duration/text()
};
```

Script d'interrogation du concept complexe « TemporalIndicator »

```
declare function drew:chatTemporalIndicator($doc as xs:string) as node()*
{
    let $temporalIndicatorBegintimestamps := drew:chatTemporalIndicatorBeginTimestamp($doc)
    let $temporalIndicatorDuration := drew:chatTemporalIndicatorDuration($doc)
    for $i at $c in $temporalIndicatorBegintimestamps
    return
    <TemporalIndicator>
    {
    <BeginTimestamp>{$temporalIndicatorBegintimestamps[$c]}</BeginTimestamp>,
    <Duration>{$temporalIndicatorDuration[$c]}</Duration>
    }
    </TemporalIndicator>
};
```

Script d'interrogation du concept simple « Username »

```
declare function drew:chatMessageSenderUsername($doc as xs:string) as xs:string*
{
    doc($doc)//chat/../@user
};
```

Script d'interrogation du concept complexe « Sender »

```
declare function drew:chatMessageSender($doc as xs:string) as node()*
{
    let $sendersUsernames := drew:chatMessageSenderUsername($doc)
    for $i in $sendersUsernames
    return
    <Sender>
    {
    <ParticipantUsername>{$i}</ParticipantUsername>
    }
    </Sender>
};
```

Script d'interrogation du concept simple « MessageContent »

```
declare function drew:chatMessageContent($doc as xs:string) as xs:string*
{
    let $msg := doc($doc)//chat/text
    for $i in $msg
    return
    if (empty($i/text()))
    then ""
    else
    $i/text()
};
```

Script d'interrogation du concept complexe « ChatMessage »

```
declare function drew:chatMessage($doc as xs:string) as element()*
{
    let $messagesContents := drew:chatMessageContent($doc)
    for $i at $j in $messagesContents
    return
    <ChatMessage>
    {<MessageContent>
    {
    $messagesContents[$j]
    }
    </MessageContent>
    }
    </ChatMessage>
};
```

Figure 61 Script d'interrogation des concepts composant le concept « Interaction de chat »

Script d'extraction du concept complexe « ChatInteraction »

```
declare function drew:chatInteraction($doc as xs:string) as node()*
{
    let $chatMessagesSenders := drew:chatMessageSender($doc), $chatTemporalIndicator :=
drew:chatTemporalIndicator($doc),
    $chatMessages := drew:chatMessage($doc)
    for $i at $c in $chatMessages
    return
    <ChatInteraction>
    {
    $chatMessagesSenders[$c],
    $chatTemporalIndicator[$c],
    $chatMessages[$c]
    }
    </ChatInteraction>
};
```

Figure 62 Script d'extraction relatif au concept complexe "Interaction de chat"

Script de filtrage sur la taille du message

```
declare function drew:chatInteractionMessageLengthGE($chatInteractions, $length) as node()*
{
    for $i in $chatInteractions
    return
    if(fn:string-length($i/ChatMessage/MessageContent/text())>=$length)
    then
    $i
    else
    ()
};
```

Script de filtrage sur l'utilisateur ayant envoyé le message

```
declare function drew:chatInteractionByUser($chatInteractions as node()*, $user as
xs:string) as node()*
{
    for $i at $c in $chatInteractions
    return
    if($i/Sender/ParticipantUsername/text()[.=$user])
    then
    $i
    else
    ()
};
```

Script de filtrage gardant le message contenant une chaîne de caractères donnée

```
declare function drew:chatInteractionMessageContainsString($chatInteractions, $expression)
as node()*
{
    for $i in $chatInteractions
    return
    if(fn:contains($i/ChatMessage/MessageContent/text(), $expression))
    then
    $i
    else
    ()
};
```

Figure 63 Script de filtrage sur la taille d'un message envoyé, l'utilisateur l'ayant envoyé, et filtrage par mot clé

6.9 Synthèse

Le modèle opérationnel présenté dans cette section, contribue, à côté du modèle sémantique des concepts interrogeables, à apporter une réponse aux contraintes relatives au (1) couplage fort entre les environnements d'apprentissage et les outils d'analyse, et (2) à la diversité de représentation des données des corpus et l'absence d'une représentation standard. Le modèle sémantique permet au chercheur d'identifier, parmi les concepts définis, ceux qui sont pertinents pour l'interrogation des ressources contenues dans un corpus. Le modèle opérationnel, en définissant six types d'opérations, donne les moyens techniques permettant d'aligner, extraire, convertir et formater les données à analyser. Ce modèle sémantique représente pour nous le moyen de construire les extractions et conversions nécessaires pour réaliser l'interface entre un corpus et un outil d'analyse. Dans les projets existants, pour qu'un corpus puisse être partagé et éventuellement analysé, des conversions préalables doivent être réalisées sur les données pour les mettre aux bons formats préconisés par ces projets. Dans notre cas, la conversion est nécessaire uniquement sur la partie du corpus qu'un chercheur souhaite analyser. Par ailleurs, notre démarche permet la réutilisation de scripts dans la construction d'autres scripts plus complexes. Ce troisième modèle contribue à la flexibilité de l'approche « Proxyma » en permettant de définir les scripts selon les besoins d'analyse, et à l'utilisabilité de l'approche en permettant la réutilisation de scripts prédéfinis.

Chapitre 7 : Evolutivité du modèle

7.2 Introduction

Ce chapitre aborde la question de l'évolution des modèles de l'approche « Proxyma » permettant son adaptation aux besoins d'analyse évolutifs des chercheurs. Nous commençons par l'explicitation du besoin d'évolution. Nous présentons par la suite les possibilités d'évolution du modèle sémantique des concepts interrogeables et du modèle opérationnel.

7.3 Le besoin d'évolution

Comme nous l'avons déjà noté dans le chapitre 5, l'approche « Proxyma » que nous proposons est incrémentale et participative. L'approche « Proxyma » définit trois modèles : le modèle de corpus, le modèle sémantique des concepts interrogeables des corpus, et le modèle

opérationnel de l'interrogation des corpus. Le premier modèle sert à la documentation des corpus partagés. Les chercheurs disposent généralement de corpus sous forme de ressources éparpillées et rarement documentées. Ils doivent donc consacrer du temps pour fournir les données descriptives qui leur sont demandées. Pour construire un corpus, notre approche consiste à récupérer les ressources du chercheur sans lui demander de les structurer préalablement en suivant un format particulier, mais en lui demandant de fournir des données pour décrire le corpus et ses composants.

Le modèle de corpus que nous proposons est simple et permet de décrire l'essentiel d'un corpus. Nous n'avons actuellement pas identifié de besoin spécifique pour étudier l'évolution de ce modèle. Ce modèle est un moyen permettant à un chercheur d'explorer un corpus qu'il ne connait pas. Si un chercheur dispose de données descriptives non explicitement exprimées dans les métadonnées de description proposées par le modèle de corpus, il a toujours la possibilité de les exprimer dans une ressource de description liée au corpus.

Les deux autres modèles peuvent par contre susciter des besoins d'évolution. En effet, la définition du modèle sémantique des concepts interrogeables d'un corpus, ainsi que le modèle opérationnel définissant les scripts d'interrogation des corpus évoluent en fonction des besoins d'analyse. Le modèle sémantique des concepts interrogeables définit les concepts qui représentent les données contenues dans les corpus partagés et qui sont éventuellement utiles à interroger en vue d'une analyse. Nous avons expliqué que la définition de ce modèle est incrémentale et participative. Les concepts définis peuvent être enrichis au fur et à mesure de l'évolution des besoins et de la richesse de la base de corpus. Ce modèle sémantique contient les concepts jugés génériques et non spécifiques à un environnement d'apprentissage ou un outil d'analyse particuliers. Il reste cependant possible que des concepts spécifiques existent dans les données d'un corpus et qu'il soit intéressant de les analyser. Dans ce cas, nous considérons qu'un enrichissement adapté du modèle peut se faire sous forme d'un module ontologique pouvant se greffer au modèle générique dans le cadre d'un contexte d'analyse particulier. Afin d'éviter des redondances dans les concepts du modèle sémantique, un dictionnaire de synonymes peut être utilisé pour détecter l'existence d'un concept dans l'ontologie et qu'un chercheur essaye d'ajouter de nouveau en utilisant un autre terme.

Le modèle opérationnel décrit six types de scripts permettant d'extraire, convertir et formater les données d'un corpus à analyser à l'aide d'un outil d'analyse. Suivant les besoins

d'analyse d'un chercheur, celui-ci définit les scripts nécessaires à la préparation des données. Ces scripts sont liés aux formalismes de représentation des données du corpus et des données attendus par un outil d'analyse particulier. La définition de ces scripts se fait d'une manière incrémentale et un chercheur n'est pas censé définir des scripts pour extraire des données du corpus qui ne lui sont pas utiles pour ses analyses. Ceci justifie le besoin d'évolution de ce modèle opérationnel au fur et à mesure de l'évolution des besoins. On peut imaginer que dans certains cas, le modèle sémantique et le modèle opérationnel évoluent d'une manière parallèle. En effet, le chercheur ajoute des concepts qui lui sont utiles dans son travail d'analyse et qui n'existaient pas dans le modèle sémantique défini par l'ontologie et définit les scripts correspondants du modèle opérationnel.

7.4 Evolution du modèle sémantique

L'évolution du modèle sémantique des concepts interrogeables des corpus se fait par la définition de nouveaux concepts et leur intégration à l'ontologie. Cette intégration peut être directement effectuée dans l'ontologie centrale partagée entre les chercheurs dans le cas de l'ajout d'un concept générique ou être définie sous forme de lien entre un module ontologique complémentaire et l'ontologie centrale dans le cas de l'ajout d'un concept spécifique à une situation et/ou des environnements d'apprentissage ou d'analyse particuliers.

Nous distinguons donc deux types d'évolutions de l'ontologie. Le premier concerne l'ajout d'un concept générique, alors que le deuxième traite de l'ajout d'un concept spécifique.

7.4.1 Ajout d'un concept générique

L'ajout d'un concept générique intervient lorsqu'un chercheur identifie un concept qu'il considère générique et utile pour ses analyses et auquel aucun concept de l'ontologie centrale ne correspond. Il est possible, dans ce cas, d'enrichir l'ontologie pour ajouter ce concept générique qui pourrait éventuellement être utile pour d'autres chercheurs. Le concept ajouté peut être un concept simple ou un concept complexe. Dans le cas d'un concept simple, ce dernier peut être un constituant d'un concept complexe précédemment défini dans l'ontologie. Si c'est un concept complexe, alors il peut être défini comme l'agrégation d'autres concepts définis dans l'ontologie ou ajoutés par le chercheur. Comme déjà présenté dans la section 5.6, les concepts du modèle sémantique sont définis sous des catégories permettant de les classer

d'une manière thématique. Nous avons expliqué qu'il est envisageable d'ajouter de nouvelles catégories de concepts au besoin. L'ajout d'un concept générique peut donc concerner soit une catégorie de concepts existante que l'on enrichit, soit une nouvelle catégorie ajoutée à l'ontologie centrale définissant le modèle sémantique. Les deux paragraphes suivants introduisent ces deux cas de figure.

7.4.1.1 Ajout du concept à une catégorie existante

Un chercheur, souhaitant analyser des données d'un corpus, peut identifier un concept générique utile pour ses analyses et qui n'existe pas dans l'ontologie centrale. Pour l'ajout d'un tel concept, le chercheur peut identifier parmi les catégories définies dans l'ontologie, la catégorie sous laquelle « son » concept peut être classé. Il est à noter qu'il n'est pas conseillé de multiplier les catégories des concepts. Cela évitera de complexifier le parcours des concepts de l'ontologie. Si une catégorie existe et qu'elle est d'un point de vue sémantique assez large pour accueillir un nouveau concept, celui-ci peut être ajouté à cette catégorie sans recourir à la création d'une nouvelle catégorie. L'ajout du concept se fait par sa création à l'aide de l'éditeur de l'ontologie utilisé (Protégé[11] dans notre cas), par sa description par une définition textuelle. Le concept ajouté peut être lié à d'autres concepts au moyen de la relation d'agrégation.

7.4.1.2 Ajout du concept à une nouvelle catégorie

Lorsqu'un chercheur souhaite ajouter un concept générique à l'ontologie et qu'il ne trouve aucune catégorie convenable pour classer ce concept, il lui est possible de définir une nouvelle catégorie permettant d'intégrer ce concept. Le chercheur peut définir une nouvelle catégorie en lui affectant un nom pertinent et en lui donnant une définition textuelle permettant de décrire la sémantique choisie pour la catégorie. Une fois la nouvelle catégorie définie, l'ajout d'un nouveau concept à cette catégorie se fait de la même manière que l'ajout d'un concept à une catégorie existante.

7.4.2 Ajout d'un concept spécifique

À part les concepts génériques qu'un chercheur peut trouver dans l'ontologie centrale ou qu'il lui ajoute, des concepts spécifiques peuvent aussi être utiles dans un contexte

[11] http://protege.stanford.edu

d'analyse particulier. En effet, les corpus contiennent des traces provenant de situations d'apprentissage diverses et variées et utilisant des environnements d'apprentissage différents. Suivant la problématique d'analyse d'un chercheur, celui-ci peut s'intéresser à des aspects spécifiques liés à l'environnement d'apprentissage, à la théorie d'apprentissage choisie ou à une méthodologie d'analyse particulière. La réutilisation de tels concepts spécifiques ou la capitalisation des analyses utilisant ces concepts peuvent être utiles pour d'autres chercheurs, d'où le besoin d'ajouter des concepts spécifiques à l'ontologie. Nous proposons de séparer les concepts génériques, pouvant être utiles pour différents chercheurs dans l'objectif d'étudier différentes questions de recherche, et les concepts spécifiques, dont le partage peut intéresser moins de chercheurs. Ceci permettra de diminuer les efforts de maintenance et d'évolutivité de l'ontologie. L'ajout des concepts spécifiques se fait d'une manière modulaire qui vient compléter l'ontologie centrale. Un module ontologique contient un ensemble de concepts spécifiques qui peuvent être liés à un type d'environnement d'apprentissage particulier, à une théorie d'apprentissage ou à une méthodologie d'analyse particulière. De nouveaux modules ontologiques peuvent donc être définis par les chercheurs suivant leurs besoins d'analyse particuliers. La définition d'un nouveau module ontologique revient à créer une nouvelle catégorie de concepts et à définir les concepts spécifiques que l'on classe sous cette catégorie. Le module ontologique est créé sous forme d'une ontologie OWL à part que l'on lie à l'ontologie centrale à l'aide de la commande « import » définie dans le standard OWL. La catégorie de concept définissant un module ontologique peut être classée, au besoin, comme une sous catégorie permettant de spécifier une catégorie existante définie dans l'ontologie centrale. Comme pour l'enrichissement de l'ontologie par un concept générique, un chercheur peut enrichir un module ontologique précédemment défini par un concept spécifique. L'ajout d'un nouveau concept spécifique peut donc se faire en enrichissant un module existant ou en créant un nouveau module ontologique.

7.4.2.1 Ajout du concept à un module existant

L'ajout d'un nouveau concept spécifique à un module existant suppose l'existence d'un module ontologique définissant une catégorie de concepts pouvant intégrer le concept. Ce module peut avoir été défini par le même chercheur pour des analyses antérieures ou par un autre chercheur qui s'intéresse à des problématiques connexes. L'ajout du concept se fait de la même façon que l'ajout d'un concept générique. En effet, le concept est décrit par un nom

significatif et une définition textuelle, et lié, au besoin, à d'autres concepts à l'aide de la relation d'agrégation.

7.4.2.2 Ajout du concept à un nouveau module

Un chercheur peut avoir besoin de créer un nouveau module ontologique qui contiendra les concepts spécifiques qui lui sont utiles pour ses analyses. Deux cas de figures peuvent se présenter : (1) le chercheur parcourt les modules ontologiques partagés par d'autres chercheurs et ne trouve pas où intégrer ses concepts spécifiques ; (2) le chercheur ne souhaite pas passer du temps pour consulter les modules précédemment définis et crée un nouveau module ontologique pour contenir ses concepts avec le risque de « redondance ». Dans le second cas, des alignements sémantiques entre les concepts de deux modules ontologiques peuvent être envisagés. L'alignement d'ontologies est un champ de recherche très actif, et des états de l'art ont été réalisés à ce propos (Euzenat et al., 2004) (Kalfoglou et Schorlemmer, 2003). Différentes techniques d'alignement peuvent être utilisées séparément ou de manière complémentaire, telles que les méthodes terminologiques et structurelles. Dans notre cas, nous considérons que l'alignement doit être fait par les utilisateurs qui identifient des correspondances entre les concepts qui peuvent être alignés grâce aux définitions associées à ceux-ci. Un tel alignement peut être facilité par l'application d'une méthode terminologique préalable se basant sur un dictionnaire permettant de retrouver des correspondances entre les concepts sur la base de leur sémantique. Les résultats des alignements terminologiques peuvent donc être validés ou refusés par l'utilisateur.

La création d'un nouveau module ontologique revient à créer une mini-ontologie OWL qui contiendra les différents concepts spécifiques proposés par le chercheur. Pour pouvoir travailler sur l'ontologie centrale enrichie par un ou plusieurs modules ontologiques spécifiques, le chercheur travaille sur une version de l'ontologie centrale dans laquelle les modules supplémentaires sont importés.

Il est à noter qu'un concept simple peut devenir complexe si un chercheur juge utile de le décomposer en un ensemble de concepts de niveau de granularité plus bas. Le traitement de ce cas est le même si le concept à ajouter/décomposer est générique ou spécifique. Les nouveaux concepts sont ajoutés à l'ontologie, et liés au concept simple qui devient complexe par la relation d'agrégation.

7.5 Evolution du modèle opérationnel

Le modèle opérationnel, ayant pour fonction de définir les mécanismes d'interrogation des corpus en se basant sur les concepts définis dans l'ontologie partagée, évolue en fonction des besoins d'analyse des chercheurs. En effet, il est évident que l'évolution du modèle sémantique par l'ajout de nouveaux concepts par un chercheur donne suite à une évolution du modèle opérationnel permettant d'interroger ces nouveaux concepts. Cependant, l'évolution du modèle opérationnel peut être indépendante de l'évolution du modèle sémantique. En effet, trois autres cas de figure peuvent se présenter. Premièrement, le partage, dans un corpus, de données ayant un nouveau format de représentation, nécessite la définition des scripts correspondants. En effet, le chercheur qui met à disposition ses données dans un corpus peut ne pas avoir besoin de définir de nouveaux concepts si ceux-ci sont déjà définis dans l'ontologie ou les modules ontologiques existants, mais doit définir les scripts d'extraction relatifs aux concepts utiles pour son analyse et liés au nouveau format. Deuxièmement, l'évolution peut être liée au besoin de formater les données extraites suivant un format de données d'entrée spécifique à un nouvel outil d'analyse. Si le chercheur utilise un nouvel outil d'analyse, les scripts permettant de préparer les données d'entrée de cet outil doivent donc être définis. Enfin, il est possible qu'un chercheur ait besoin de définir un script de conversion de types de données dont il a besoin et qui n'est pas défini dans le modèle opérationnel.

7.5.1 Évolution due à celle du modèle sémantique

Pour répondre à des besoins d'analyse, un chercheur peut ajouter de nouveau concepts génériques ou spécifiques à l'ontologie centrale ou les modules ontologiques. Pour permettre l'interrogation de ces nouveaux concepts, le chercheur définit les scripts correspondants permettant d'extraire les données relatives à ces concepts dans les ressources partagées dans un corpus. Si le concept ajouté est un concept simple alors le chercheur définit les scripts d'interrogation qui lui sont relatifs et met à jour les scripts relatifs aux concepts complexes qui lui sont liés par la relation d'agrégation. L'évolution concerne les scripts définis dans le chapitre 6 : script d'interrogation de concept, script d'extraction, script de filtrage, script de formatage, et script de fusion.

7.5.2 Évolution liée à un nouveau format de traces

Lorsqu'un chercheur construit un nouveau corpus et y partage des ressources de traces, brutes ou enrichies, ayant des formats nouveaux pour lesquels les scripts d'extraction correspondants n'ont pas été définis, le modèle opérationnel doit évoluer. Le chercheur définit donc les scripts correspondants lui permettant d'extraire les données relatives aux concepts utiles pour l'analyse à partir des ressources. Le chercheur définit les différents types de scripts dont il a besoin. Comme présenté dans le chapitre 6, les scripts liés au format des données interrogés sont : script d'interrogation de concept, script d'extraction, script de filtrage, script de formatage, et script de fusion.

7.5.3 Évolution liée à un nouveau format d'entrée d'outil d'analyse

Si un chercheur souhaite utiliser un nouvel outil d'analyse ayant un format de données d'entrée particulier pour lequel les scripts relatifs n'ont pas été définis, il doit ajouter ces scripts au modèle opérationnel. Comme déjà présenté dans le chapitre 6, les scripts du modèle opérationnel qui sont liés au format d'entrée de l'outil d'analyse sont : script d'extraction, script de filtrage, script de formatage, et script de fusion.

7.5.4 Évolution par l'ajout de scripts génériques de conversion

Le quatrième cas de figure d'évolution du modèle opérationnel concerne les scripts génériques de conversion de types de données. Ces scripts servent à convertir une donnée d'un type de données en entrée en un type de données en sortie. Si le chercheur a besoin de définir un tel script qui lui servira dans ses scripts d'extraction et qui est assez générique pour être réutilisé par d'autres utilisateurs, il peut enrichir le modèle opérationnel en ajoutant un script de conversion de types de données.

7.6 Coût de l'évolution des modèles

Afin d'assurer une utilisabilité importante de l'approche « Proxyma » que nous proposons, il convient d'étudier le coût de l'évolution du modèle sémantique et du modèle opérationnel et de mesurer la part de travail qui incombe au chercheur souhaitant interroger un corpus.

En ce qui concerne l'évolution du modèle sémantique, celle-ci ne nécessite pas beaucoup de temps ni de compétences informatiques approfondies. En effet, pour proposer l'ajout d'un nouveau concept, le chercheur a besoin de consulter les catégories et les concepts existants, d'intégrer son nouveau concept sous la catégorie qui lui convient et de lui associer une définition textuelle. Nous utilisons actuellement l'éditeur d'ontologie Protégé pour manipuler l'ontologie. Protégé est facile d'utilisation et peut être utilisé par des chercheurs non informaticiens. Mais dans le cadre du développement d'une application Web permettant la manipulation de la plateforme « Beatcorp », nous pouvons imaginer le développement d'une interface graphique permettant une manipulation plus simple de l'ontologie.

En ce qui concerne le modèle opérationnel, le chercheur qui souhaite interroger un corpus peut chercher si les scripts nécessaires à l'extraction des données existent afin de les réutiliser. Sinon, le chercheur doit définir lui-même les différents types de scripts dont il aura besoin. Afin d'optimiser son temps de travail, le chercheur peut définir uniquement les scripts lui permettant d'extraire les données dont il a besoin et éviter ainsi de faire un travail systématique d'extraction de données dont il n'a pas besoin dans l'immédiat. L'idée est d'écrire les scripts quand il en a besoin. Dans l'état actuel du travail réalisé comme « preuve de concept », les scripts sont écrits en XQuery (XQuery, 2010) ce qui nécessite un minimum de connaissances informatiques en programmation. La charge de travail du chercheur est proportionnelle à ses besoins en termes d'extraction, de conversion, de filtrage et de mise en forme des données des corpus. La disponibilité de scripts développés par d'autres chercheurs et accessibles dans la plateforme permet de diminuer considérablement le temps d'écriture des scripts par un chercheur qui peut s'inspirer du travail réalisé précédemment. Ceci n'étant pas évident pour les chercheurs non informaticiens, nous étudions dans les perspectives de ce travail la possibilité de fabriquer des scripts (semi-)automatiquement ainsi que le développement d'un langage graphique de création de scripts.

7.7 Synthèse

La démarche que nous proposons permet la définition incrémentale et participative des concepts de l'ontologie permettant l'interrogation des corpus. L'ontologie évolue en fonction des besoins d'analyse des chercheurs sans avoir besoin de faire un travail préalable exhaustif, qui sera sans doute imparfait, de recensement des concepts utiles pour l'interrogation des données contenues dans les corpus de traces d'interactions contextualisées. Les modèles

sémantique et opérationnel de l'approche « Proxyma » évoluent donc au fur et à mesure des besoins d'analyse des chercheurs pour permettre une définition évolutive des concepts nécessaires aux analyses.

Chapitre 8 : Architecture de « Beatcorp », une plateforme de partage basée sur l'approche « Proxyma »

8.2 Introduction

Dans ce chapitre, nous proposons l'architecture de la plateforme « Beatcorp » de partage et d'analyse de corpus de traces. Cette architecture est basée sur l'approche « Proxyma » et ses trois modèles. Nous commençons par une introduction brève des standards de représentation de données XML, RDF, et OWL, ainsi que du langage XQuery d'interrogation de documents XML. Nous présentons ensuite les cinq composants de l'architecture de la plateforme que nous proposons, et soulignons la possibilité de réutiliser une plateforme de gestion de base de données existante. Nous démontrons par la suite l'utilité de la plateforme aussi bien pour un travail individuel que pour un travail centralisé. Nous décrivons par la suite trois exemples de scénarios d'utilisation de la plateforme « Beatcorp ». Enfin, nous montrons que l'approche « Proxyma » sous-tendant la plateforme répond bien aux critères identifiés dans le chapitre 3 pour traiter nos questions de recherche.

8.3 Standards de représentation et d'interrogation de données

Cette section est consacrée à une présentation brève des langages standards de représentation de données XML, RDF et OWL, ainsi que du langage XQuery d'interrogation de ressources XML. Ces langages sont des recommandations du W3C (World Wide Web Consortium) et sont de plus en plus utilisés dans le contexte des applications du Web et du Web sémantique.

8.3.1 XML

XML (eXtensible Markup Language) (XML, 2008) est un language de balisage extensible. Ce langage simple et flexible est conçu pour faciliter l'interopérabilité et l'échange de contenus complexes entre systèmes d'informations hétérogènes sur le Web ou ailleurs. Un document XML est organisé dans une structure arborescente. La structure d'un document XML et les types des données qu'il contient peuvent être exprimés dans une DTD (Datatype Definition (DTD, 2008) ou définition de types de données). Par ailleurs, le schéma XML (XSD, 2004) est une méthode plus élaborée pour la structuration et la définition des types de données d'un document XML. Un schéma XML, contrairement à une DTD, est exprimé en langage XML. XSD offre une gestion plus fine de la structure d'un document XML, en

permettant, entre autres, de définir de nouveaux types de données, et d'exprimer des contraintes sur le séquencement des contenus. Un document XML est un arbre dont les feuilles sont des données sous forme de chaînes de caractères. Un document XML contient des éléments représentant les nœuds de l'arbre qui sont représentés par des balises portant un label (tag), un élément pouvant être décrit par des attributs. Un attribut possède un nom et une valeur. Chaque nœud peut avoir des fils. L'utilisation de XML donne la possibilité d'exploiter un panel très riche d'outils d'édition et d'interrogation des données. XML est de plus en plus utilisé dans la représentation des données par des standards. Par exemple, (IEEE 1484.12.3, 2003) offre une définition de l'application du standard LOM sous forme d'un schéma XML. Les standards offerts par l'IMS Global Learning Consortium (IMS GLC, 1997), tel que le standard IMS LD (IMS LD, 2003) de modélisation pédagogique, proposent des schémas XML. Par ailleurs, beaucoup de travaux dans le domaine des EIAH utilisent le langage XML pour représenter les traces d'activité générés par les environnements d'apprentissage (mulce-struct (Reffay et al., 2008), tutor message format (Tutor Message Format, 2013), common format (Martínez et al., 2005), CAM (Wolpers et al., 2007), etc.).

8.3.2 RDF

RDF (Resource Description Framework ou Cadre de description de ressource) (RDF, 2004) est un formalisme de description de ressources Web. RDF est essentiellement conçu pour représenter des métadonnées sur des ressources Web, comme le titre, l'auteur, etc. Le concept de ressource Web peut être généralisé pour décrire tout objet identifiable sur le Web, par exemple, des informations sur les préférences d'un utilisateur Web. RDF est destiné aux situations où les descriptions sont traitées automatiquement par des applications informatiques et non destinées aux seuls humains. RDF est un modèle de données indépendant du domaine. Un document RDF est un ensemble de triplets (sujet, prédicat, objet). Un triplet RDF est composé de deux nœuds liés par un arc orienté. Le premier nœud représente le sujet décrit, l'arc orienté est le prédicat de la propriété décrite, et le deuxième nœud est l'objet lié au sujet par le prédicat. RDF peut être exprimé par une syntaxe XML appelée XML/RDF mais d'autres syntaxes existent. Pour décrire une ressource Web, il peut être utile de définir un vocabulaire qui définit les propriétés utiles pour la description. Le vocabulaire d'un document RDF peut être décrit en RDF schema RDFS (RDFS, 2004), le langage de description de vocabulaire RDF. Il est exprimé en RDF et permet la définition des classes des ressources utilisées et les propriétés utiles pour leur description. Un exemple de vocabulaire contrôlé

RDF est le célèbre Dublin Core (DCMI, 1994) qui définit un ensemble de propriétés/métadonnées pour la description de ressources.

8.3.3 OWL

Le langage OWL (Ontology Web Language ou Langage ontologique pour le Web) (OWL, 2004) est un langage de définition d'ontologies Web basé sur les recherches dans le domaine de la logique de description. L'ontologie, terme originalement philosophique, désigne la science qui décrit les types d'entités dans le monde et leurs relations. OWL est basé sur les standards RDF et RDF-Schema en ajoutant du vocabulaire de description des propriétés et des classes. OWL formalise un domaine par la définition des classes et des propriétés de ces classes. Les propriétés peuvent être des propriétés d'objets, permettant d'exprimer des relations entre classes (p. ex. une personne possède une voiture), ou des propriétés de types de données permettant d'exprimer une relation entre une classe et son attribut (p. ex. une personne et son nom). Il permet de définir des « individus » instances des classes et d'affirmer des propriétés à leur sujet, et de raisonner sur ces classes et individus. Le langage OWL est très utilisé dans la représentation des ontologies du fait de l'expressivité offerte par ce langage. Des moteurs de raisonnements, tel que Fact++ (Tsarkov et Horrocks, 2006), sont accessibles pour raisonner sur les faits qui composent l'ontologie. Un éditeur d'ontologie nommé Protégé (Protégé, 2013) est très utilisé et offre la possibilité de construire, visualiser, et manipuler des ontologies. OWL et Protégé représentent une bonne infrastructure pour la construction et l'exploitation d'ontologies.

8.3.4 XQuery

XQuery (XML Query Language ou Langage d'interrogation pour XML) (XQuery, 2010) est une recommandation du W3C. XQuery est pour XML ce que SQL est pour les bases de données relationnelles. Ce langage est basé sur le langage XPath (XPath, 2013) qui est un langage de navigation à travers les éléments et attributs dans un document XML. XQuery permet d'extraire des données à partir d'un document ou d'une collection de documents XML. XQuery dispose de constructions puissantes telle l'expression FLWOR (F : for, L : let, W : where, O : order by, R :return). Cette expression permet d'itérer sur des séquences (clause *for*), créer des liaisons temporaires variable-valeur (clause *let*), filtrer des séquences (clause *where*), trier des séquences (clause *order by*), et de structurer le résultat attendu sous forme d'un ensemble de fragments XML (clause *return*).

8.4 Architecture de la plateforme de partage « Beatcorp »

L'approche « Proxyma » est notre proposition pour offrir une solution permettant le partage entre chercheurs de corpus de traces d'interactions enrichies en vue de les analyser et les enrichir davantage. Pour exploiter cette approche, nous proposons l'architecture de la plateforme « Beatcorp » basée sur l'approche « Proxyma ». La Figure 64 ci-dessous illustre cette architecture.

L'architecture comporte (1) une base de corpus permettant le stockage des corpus, (2) une ontologie permettant la formalisation des trois modèles de l'approche, (3) une base des scripts qui stocke les scripts relatifs aux six types d'opérations définies par le modèle opérationnel et permettant l'exploitation des ressources des corpus, (4) un moteur de gestion permettant la manipulation des différents composants de la plateforme, et (5) une application Web jouant le rôle d'interface entre le chercheur et les autres composants.

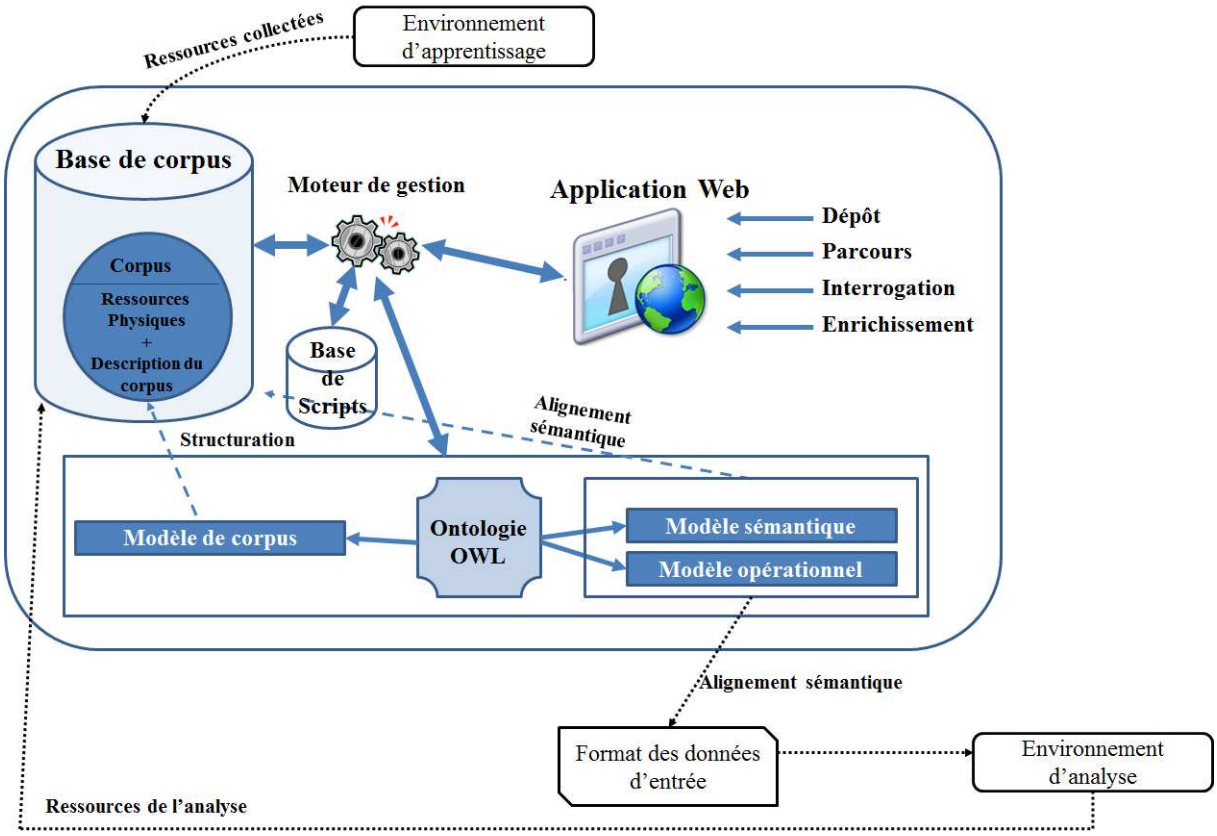


Figure 64 Architecture de la plateforme « Beatcorp » de partage de corpus

8.4.1 Base de corpus

La base de corpus permet le stockage des corpus partagés par des chercheurs pour des chercheurs. Comme le spécifie le modèle de corpus que nous proposons dans le chapitre 4, un corpus partagé peut être un corpus initial dont les ressources collectées proviennent d'une expérimentation d'apprentissage et des éventuelles analyses réalisées sur ce corpus avant sa construction et son partage dans la plateforme. Un corpus partagé peut aussi être un corpus d'analyse permettant de stocker et décrire des ressources relatives à des analyses réalisées sur un ou plusieurs corpus originaux ou d'analyse déjà partagés dans la plateforme dans le but de répondre à une question de recherche particulière. Un corpus partagé dans cette base est donc équivalent à un ensemble de ressources physiques et d'une description du corpus et de son contenu. Les types des ressources pouvant être partagées dans un corpus sont présentés dans le paragraphe 4.3.3. La description d'un corpus se fait selon le modèle de description de corpus présenté dans le paragraphe 4.3.5.

La base de corpus permet le stockage des données fournies par des chercheurs pour les partager avec d'autres chercheurs. Pour analyser ces corpus en utilisant différents outils d'analyse, il convient de pouvoir interroger la base et la mettre à jour en ajoutant de nouvelles ressources aux corpus et en modifiant les descriptions des corpus. La manipulation de la base des corpus se fait par le biais du moteur de gestion qui sera présenté dans la suite.

8.4.2 Ontologie : définition formelle des modèles de l'approche

Le deuxième composant de la plateforme « Beatcorp » est l'ontologie qui définit d'une manière formelle les trois modèles de l'approche « Proxyma » : le modèle de corpus, le modèle sémantique, et le modèle opérationnel. Cette ontologie est exprimée en OWL.

Nous utilisons le langage d'ontologie OWL et l'éditeur d'ontologies Protégé pour construire l'ontologie qui formalise nos modèles. Le modèle de corpus présenté dans le chapitre 4 est conceptualisé par l'ensemble des classes et des propriétés utiles pour la description d'un corpus partagé dans la plateforme. La Figure 65 ci-dessous illustre une partie du modèle de corpus formalisé par l'ontologie et construite en utilisant l'éditeur d'ontologies Protégé. La partie gauche de la figure illustre les noms des classes utilisées pour représenter les corpus, quant à la partie droite, elle illustre une partie des métadonnées de description des corpus. La description d'un corpus particulier se fait sous forme d'un document RDF,

exprimé dans la syntaxe RDF/XML relative au standard RDF, conforme à la conceptualisation ontologique du modèle de description de corpus présenté dans le paragraphe 4.3.5.

L'utilisation des langages RDF et OWL nous permet de nous baser sur des standards du Web. Ces langages sont largement utilisés et bénéficient d'un ensemble d'outils de création, de manipulation, d'interrogation, et de raisonnement open source. Du fait que ces langages soient des standards Web, leur utilisation permet un échange et une interopérabilité plus facile avec d'éventuelles applications ultérieures. Par ailleurs, le modèle sémantique étant ouvert et extensible, des chercheurs peuvent vouloir lier l'ontologie centrale à des ontologies spécifiques définies dans le cadre de leurs recherches. La spécification de nos modèles à l'aide d'une ontologie est un choix qui permet d'élargir le spectre des possibilités de la plateforme de partage et d'analyse de corpus de traces. En effet, la conception ontologique permet de profiter des possibilités de raisonnement offertes par la logique de description sous-tendant le langage OWL. Bien que l'ontologie que nous avons définie jusque là est assez simple et que les relations entre concepts sont minimales, ce qui limite les mécanismes de raisonnement que nous pourrions utiliser, certaines relations peuvent cependant être déduites. Par exemple, nous pouvons exploiter la structure hiérarchique des concepts pour répondre à la requête : quels sont les corpus qui contiennent des traces relatives à la communication ? En utilisant un outil de raisonnement, la réponse à cette requête permettra de sélectionner tous les corpus contenant des traces liées à tous les concepts descendants de la catégorie « communication » (cf. Figure 66). La modélisation ontologique nous permet d'envisager des évolutions de notre ontologie avec plus de possibilités au niveau des raisonnements automatiques possibles grâce au langage OWL et aux moteurs de raisonnement.

Le modèle sémantique (cf. Figure 66) des concepts interrogeables des corpus présenté dans le chapitre 5 définit des concepts classés dans des catégories et représentant des informations pouvant être retrouvées dans les corpus. Comme nous l'avons déjà expliqué et illustré par des exemples dans le chapitre 5, les concepts peuvent être liés via deux types de relations. Ce modèle est ouvert et peut être enrichi pour répondre aux besoins d'analyse du chercheur.

Le modèle opérationnel définit six types d'opérations relatifs aux mécanismes opérationnels pour l'exploitation des données des corpus. L'ontologie définit les classes relatives à ces opérations ainsi que leurs relations et propriétés. Par exemple, l'opération

d'interrogation de concept est liée à la structure de la ressource interrogée, et interroge un concept défini dans le modèle opérationnel. Ce type d'opération peut interroger un concept simple ou un concept complexe. L'interrogation d'un concept complexe fait donc appel aux opérations d'interrogation des concepts simples qui composent le concept complexe. La Figure 67 et la Figure 68 ci-dessous illustrent des parties de l'ontologie représentant la conceptualisation des scripts « script d'interrogation de concept » et « script de filtrage » définis dans le modèle opérationnel. L'ontologie permet d'établir le lien entre le modèle sémantique, par la définition des concepts qui pourraient être retrouvés dans les corpus, et le modèle opérationnel, par la définition des scripts permettant l'alignement de ces concepts avec les contenus des ressources.

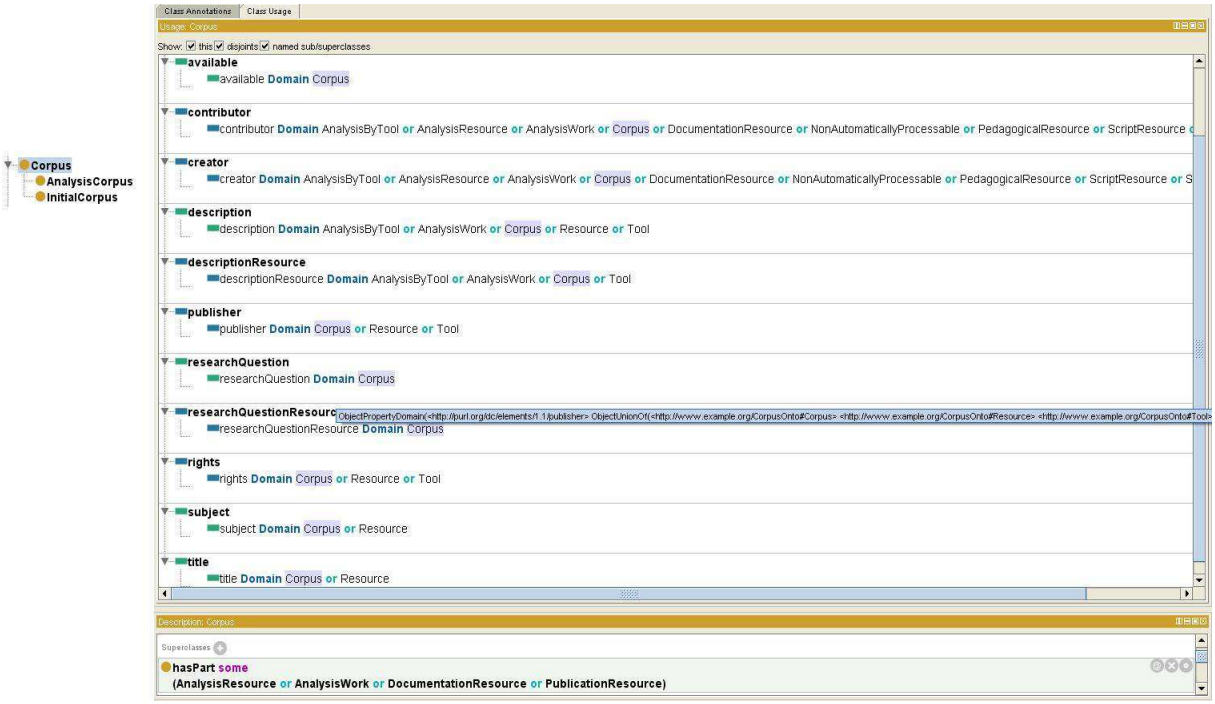


Figure 65 Extrait du modèle de corpus formalisé par l'ontologie (visualisé à l'aide de l'éditeur d'ontologie Protégé)

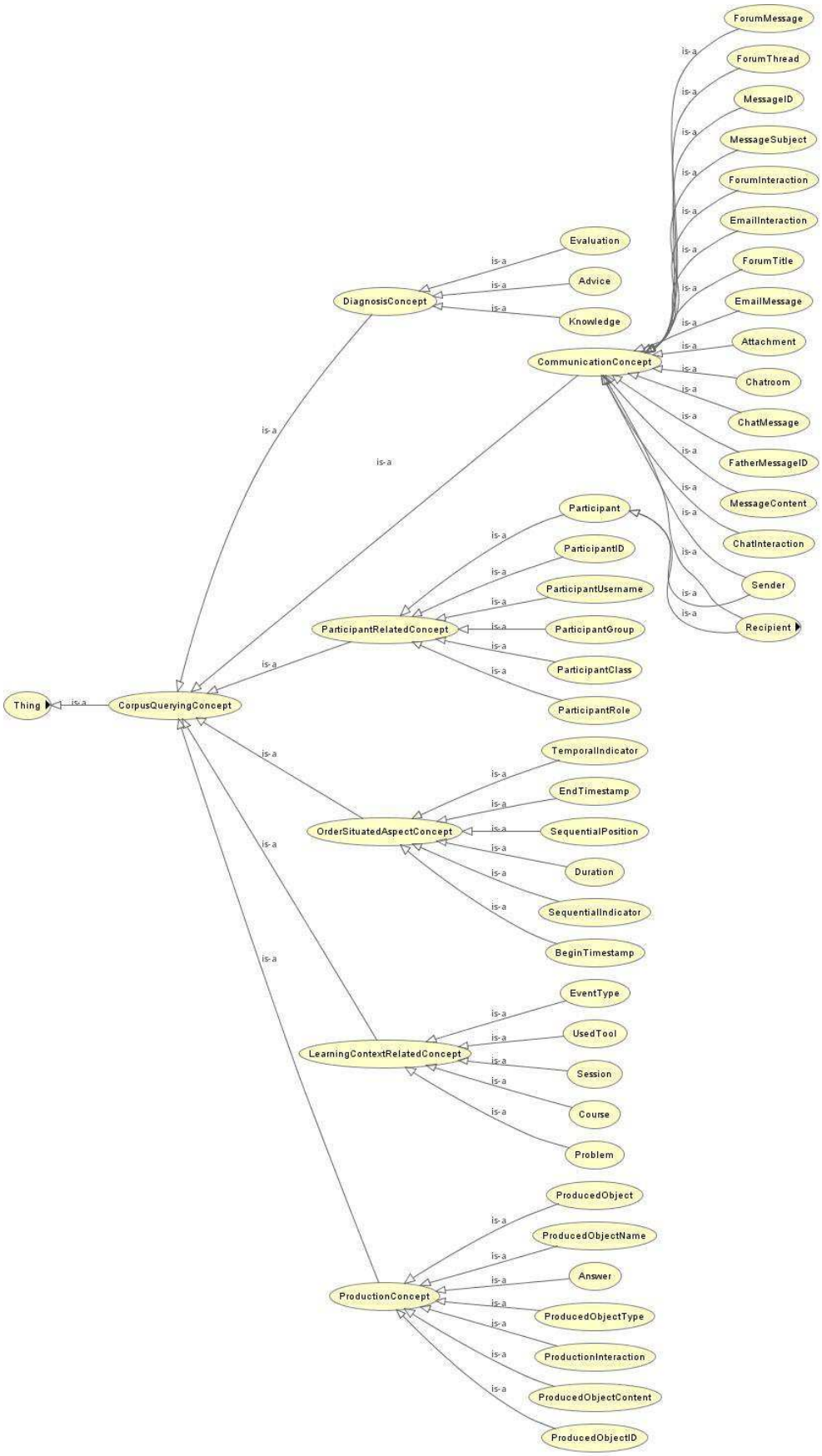


Figure 66 Extrait de l'ontologie relatif au modèle sémantique (outil de visualisation OWL Viz de l'éditeur Protégé)

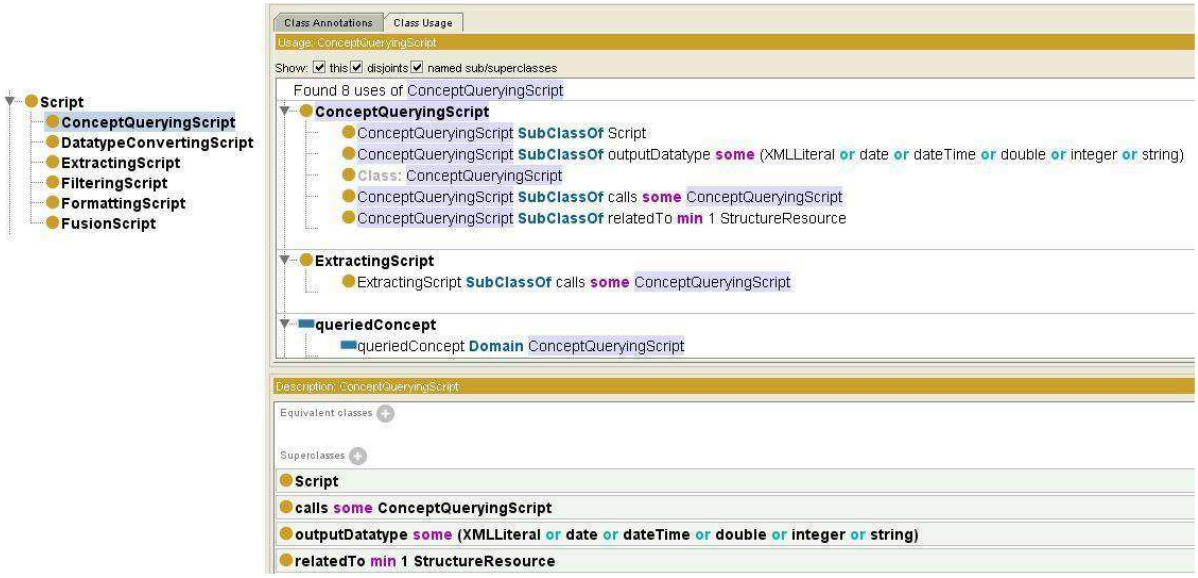


Figure 67 Extrait du modèle opérationnel formalisé par l'ontologie – formalisation du concept « script d'interrogation de concept » (visualisé à l'aide de l'éditeur d'ontologie Protégé)

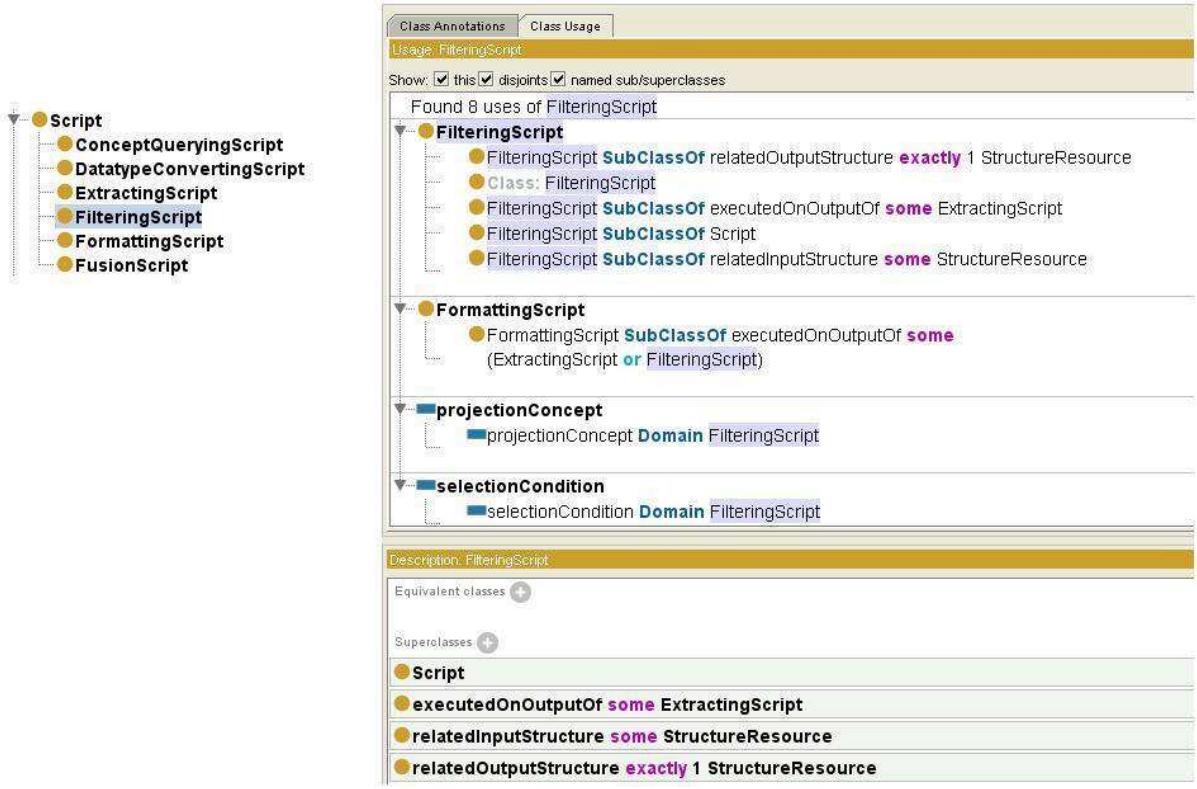


Figure 68 Extrait du modèle opérationnel formalisé par l'ontologie – formalisation du concept « script de filtrage » (visualisé à l'aide de l'éditeur d'ontologie Protégé)

8.4.3 Base de scripts

La base de scripts est le troisième composant de la plateforme « Beatcorp ». Elle contient les scripts relatifs aux opérations définies dans le modèle opérationnel permettant d'interroger les corpus afin d'analyser les données qu'ils contiennent. Cette base est construite d'une manière incrémentale pour répondre aux besoins d'analyse des chercheurs. En effet, étant conscients de la difficulté d'être exhaustif dans la définition des données pouvant exister dans des corpus et intéresser les chercheurs dans leurs analyses, et dans le développement des scripts utiles pour interroger ces données, notre objectif est de proposer une approche pouvant être rapidement opérationnelle une fois les objectifs d'analyse bien déterminés. Il est aussi à noter qu'il est très difficile, voire impossible, d'identifier d'une manière exhaustive les questions de recherche pouvant être étudiées en se basant sur un corpus donné, et par conséquent d'identifier les requêtes pouvant être pertinentes à exécuter sur un corpus. Un chercheur souhaitant analyser un corpus sélectionne l'outil d'analyse qu'il souhaite utiliser. La base de scripts est donc sollicitée par le moteur de gestion pour vérifier si les scripts nécessaires à l'extraction des données du corpus sont déjà définis auquel cas l'extraction et la conversion des données sont exécutées automatiquement. Dans le cas contraire, si des scripts manquent, ces derniers doivent être définis et liés aux concepts concernés. Les scripts sont exprimés en utilisant un langage de programmation (XQuery dans notre cas) permettant l'interrogation des données afin d'en extraire celles qui seront analysées par un outil d'analyse. Les scripts contenus dans la base de scripts doivent être interprétables par le moteur de gestion. Ce dernier exécute les scripts sur les corpus de la base de corpus et renvoie les données à analyser par un outil d'analyse. Nous avons choisi de représenter les données à interroger en langage XML. Ce langage est largement utilisé dans la représentation des traces. Par ailleurs, il est relativement facile d'exporter ou de convertir des données représentées dans un autre formalisme (p. ex. base de données, fichier CSV) dans le langage XML.

8.4.4 Moteur de gestion

Le moteur de gestion permet l'exécution des opérations nécessaires pour la manipulation de la base de corpus, la base des scripts et l'ontologie en se basant sur les trois modèles de l'approche « Proxyma ». Ce moteur est une couche intermédiaire entre l'application Web utilisée par un chercheur et les autres composants de la plateforme. Il permet d'exécuter les requêtes des utilisateurs et éventuellement de renvoyer des résultats. L'utilisation du moteur de gestion suppose l'existence d'un langage d'interrogation

permettant l'exploitation des composants de la plateforme et qui soit interprétable par ce moteur. Le moteur de gestion doit être capable d'interpréter des requêtes permettant l'insertion de nouvelles données dans la base de corpus, la suppression de données existantes, l'édition de données existantes, et l'interrogation des données stockées. Il permet également la gestion des composants ontologie et base de scripts en exécutant des requêtes d'insertion, de suppression, d'édition et d'interrogation. Nous présentons dans ce qui suit les quatre opérations de gestion assurées par le moteur de gestion pour manipuler les trois composants : base de corpus, ontologie, et base de scripts.

8.4.4.1 Insertion

L'insertion représente la première opération assurée par le moteur de gestion. Elle peut concerner les trois modèles de l'approche.

Le modèle de corpus est concerné par l'insertion lorsqu'un nouveau corpus doit être créé ou lorsque de nouvelles données doivent être insérées dans un corpus existant. Le modèle de corpus définit les métadonnées de description générale d'un corpus ainsi que les types de ressources pouvant y être partagées et les métadonnées de leur description. Le moteur de gestion utilise ce modèle formalisé par l'ontologie pour insérer des données dans la base de corpus suivant le modèle proposé. L'insertion peut donc concerner la description des métadonnées fournies par le chercheur pour décrire un corpus et ses contenus ou les ressources physiques composant un corpus.

Le modèle sémantique est le deuxième modèle de l'approche formalisé par l'ontologie. Il définit les concepts interrogeables des corpus de traces d'interactions d'apprentissage contextualisées. Nous avons déjà expliqué que le modèle sémantique est ouvert et évolutif. Il peut être enrichi par l'ajout de nouveaux concepts utiles pour les analyses d'un chercheur. Un chercheur peut donc avoir besoin de définir de nouveaux concepts qu'il identifie comme étant des concepts utiles pour ses analyses et qui ne sont pas déjà définis dans l'ontologie. Le moteur de gestion permet donc la manipulation du modèle sémantique par l'ajout de nouveaux concepts et l'expression des relations éventuelles avec des concepts existants.

Le modèle opérationnel, troisième modèle de l'approche, définit six types d'opérations utiles pour l'extraction des données des corpus à analyser par un chercheur. Le moteur de gestion s'appuie sur la formalisation ontologique du modèle opérationnel pour permettre

d'insérer, dans la base de scripts, de nouveaux scripts relatifs aux opérations qu'il définit. L'insertion peut intervenir lorsqu'un nouveau corpus est construit et qu'il contient des traces dont la structure était encore inconnue pour le système. Il convient alors d'aligner les concepts contenus dans les traces du corpus avec les concepts correspondants du modèle sémantique en définissant les différents scripts nécessaires. Ces derniers sont alors intégrés dans la base de scripts et seront désormais accessibles pour des analyses futures. L'insertion de nouveaux concepts peut aussi intervenir quand un chercheur souhaite effectuer une nouvelle analyse sur des données existant dans un corpus mais pour lesquelles les scripts (extraction, filtrage, formatage) n'ont pas encore été définis.

8.4.4.2 Suppression

La deuxième opération de gestion des composants de la plateforme est la suppression. Comme pour l'insertion, la suppression est liée aux trois modèles de l'approche proposée. La suppression peut concerner le modèle de corpus si elle consiste à supprimer un corpus, un composant d'un corpus ou des données de description d'un corpus.

La suppression peut aussi concerner le modèle sémantique. Les concepts identifiés comme étant génériques et pouvant être utiles pour différents chercheurs dans différentes analyses ne peuvent pas être supprimés. La suppression peut concerner les concepts spécifiques ajoutés par les chercheurs qui peuvent avoir besoin de supprimer des concepts ajoutés au modèle sémantique s'ils jugent qu'ils ne leur seront finalement pas utiles. Une telle suppression n'est possible que si le concept n'est pas utilisé par d'autres chercheurs. Un concept peut être complexe, et peut donc faire référence à d'autres concepts. La suppression d'un concept suppose la suppression de toute référence à ce concept. Le moteur de gestion doit permettre la vérification de cette condition. Les concepts ajoutés par un chercheur pouvant être utilisés par d'autres chercheurs utilisant la plateforme, il ne doit pas être possible de supprimer un concept référencé par un ou plusieurs scripts relatifs aux opérations du modèle opérationnel et stockés dans la base des scripts.

Enfin, la suppression peut concerner le modèle opérationnel. En effet, un script inutile ou défectueux peut être supprimé. Comme déjà présenté dans le chapitre 6, un script peut faire appel à un autre. Par exemple, un script d'interrogation d'un concept complexe fait appel aux scripts d'interrogation des concepts qui le constituent. La suppression d'un script S ne doit être possible que si aucun autre script de la base des scripts ne fait appel à S.

8.4.4.3 Édition

L'édition est la troisième opération nécessaire pour la manipulation des données de la plateforme en se basant sur les trois modèles. L'édition consiste en la modification des contenus de données existantes.

En relation avec le modèle de corpus, l'édition ne concerne que les données entrées par le chercheur lors de la construction ou de l'enrichissement d'un corpus. Elle est donc liée au modèle de description de corpus. Elle consiste à éditer les métadonnées de description d'un corpus ou de ses composants.

Le modèle sémantique peut faire l'objet d'éditions. Un chercheur peut éditer un concept en changeant le nom qui lui est attribué. Une telle édition doit faire l'objet de discussions entre les chercheurs utilisant ce concept. L'édition peut être utile si les chercheurs considèrent qu'un terme représente mieux le concept en question. L'édition peut aussi intervenir au niveau des définitions contextuelles relatives aux concepts du modèle sémantique. Une définition peut être éditée pour la rendre plus compréhensible. Par ailleurs, l'édition peut aussi concerner la définition d'un concept sous forme d'une agrégation d'autres concepts. En effet, la définition d'un concept comme l'agrégation d'un ensemble d'autres concepts peut évoluer pour répondre à de nouveaux besoins d'analyse.

Enfin, les scripts de la base des scripts relatifs aux opérations du modèle opérationnel peuvent faire l'objet d'éditions. Si le script a besoin d'être modifié pour répondre d'une manière plus appropriée au besoin d'un chercheur analyste, le moteur de gestion doit permettre cette édition. Un script pouvant être utilisés par nombre d'autres scripts, l'édition d'un script ne doit pas affecter les résultats des scripts appelants. La plateforme peut gérer ce genre de situation de différentes manières. Elle peut décider de faire confiance au chercheur et de supposer que ce dernier vérifie que l'édition d'un script n'affecte pas la cohérence d'autres scripts appelants. Elle peut interdire l'édition d'un script appelé par un autre script. Elle peut encore conserver des versions différentes de ces scripts, et lier ces versions à leurs utilisations spécifiques.

8.4.4.4 Interrogation

La quatrième opération de gestion assurée par le moteur de gestion est l'interrogation des différents composants de la plateforme. L'interrogation peut intervenir à différents niveaux d'utilisation de la plateforme.

Au niveau du modèle de corpus, l'interrogation concerne deux niveaux : le premier est relatif au modèle de description du corpus, et le deuxième aux données partagées dans les ressources physiques du corpus. L'interrogation de la description d'un corpus se fait dans le cadre d'un parcours des corpus partagés dans la base de corpus. Une telle interrogation peut-être utile pour un chercheur souhaitant avoir une idée sommaire sur les corpus partagés dans la plateforme. Cela peut permettre à un chercheur de vérifier l'existence d'un ou plusieurs corpus qui puissent l'intéresser. Le modèle de description de corpus que nous proposons nous sert donc comme schéma de base pour les requêtes. Le deuxième niveau d'interrogation concerne les données stockées dans les ressources physiques d'un corpus. Ces données sont interrogeables via les scripts relatifs aux opérations définies dans le modèle opérationnel. L'interrogation des ressources d'un corpus fait donc appel à des scripts stockés dans la base de scripts. Une telle interrogation suppose que les scripts nécessaires à l'extraction des données soient existants et programmés et que le moteur de gestion soit capable d'y accéder, de les exécuter, et de renvoyer le résultat de leur exécution.

L'interrogation peut aussi concerner le modèle sémantique. En effet, il peut s'avérer utile pour un chercheur d'exécuter des requêtes sur les concepts définis dans l'ontologie, que ce soit leur définition ou leurs relations. Le chercheur peut éventuellement rechercher un concept à partir de son nom. Un concept pouvant être désigné par des termes différents, on peut imaginer que l'utilisation d'un dictionnaire de synonymes des termes désignant les concepts permettrait d'élargir l'ensemble des résultats proposés au chercheur. L'interrogation peut aussi concerner les définitions données aux concepts ainsi que les relations pouvant lier les concepts (p. ex. retourner l'ensemble des concepts dont la définition sous forme d'agrégation contient un concept particulier). Les interrogations peuvent concerner les modèles sémantique et opérationnel en même temps. Par exemple, chercher les scripts contenus dans la base de scripts et qui interrogent un concept particulier. Par ailleurs, le chercheur doit pouvoir faire des recherches dans la base de scripts en se basant sur d'autres critères de recherche, comme les scripts qui font appel à un script particulier, ou les scripts liés à un format de données particulier.

8.4.5 Application Web

Le cinquième composant de la plateforme « Beatcorp » est une application Web permettant aux chercheurs d'accéder et de manipuler les autres composants de la plateforme via le moteur de gestion. L'application joue le rôle d'une interface entre le chercheur et les composants de la plateforme. Cette application doit être ergonomique et masquer, le plus possible, les aspects techniques de manipulation et d'interrogation des données. En effet, ce travail s'adresse aux chercheurs utilisant les environnements informatiques pour l'apprentissage humain dans leurs recherches. Ils peuvent donc être utilisateurs de l'outil informatique mais non informaticiens. Par exemple, un chercheur souhaitant créer un nouveau corpus ne verra pas d'une manière explicite le formalisme du modèle de corpus que nous proposons, mais ce dernier sera utilisé pour définir les champs qu'un chercheur doit saisir pour décrire le corpus et ses composants.

L'objectif étant entre autres de permettre à des chercheurs de partager des corpus, il est pertinent de rendre ces données plus visibles sur le Web en les référençant auprès des archives ouvertes, en l'occurrence en utilisant les métadonnées des corpus pour les référencer auprès des archives qui utilisent le protocole OAI-PMH (OAI-PMH, 2013).

8.5 Réutilisation d'une plateforme existante

L'architecture que nous proposons comme prototype de la plateforme « Beatcorp » et la mise en œuvre de l'approche « Proxyma » peut être partiellement gérée par une plateforme de base de données existante. Comme précédemment présenté, nous avons choisi de représenter les données partagées en utilisant des standards basés sur le langage XML de représentation des données. Nous proposons donc d'utiliser le système de gestion de base données XML native eXist-db (eXist-db, 2012), comme plateforme support pour le stockage et la manipulation des données. En effet, eXist-db offre un moteur de gestion des données, permettant l'interrogation en utilisant le langage XQuery, ainsi que des fonctionnalités de mise à jour sous forme d'extension du langage XQuery.

La base de corpus, la base des scripts ainsi que l'ontologie formalisant les trois modèles peuvent être stockées dans le système de gestion de base de données (SGBD) eXist. Le moteur de gestion utilise les outils offerts par le SGBD pour l'insertion, la mise à jour et l'interrogation.

8.6 Utilisabilité de la plateforme « Beatcorp » : données privées et données partagées

La plateforme de partage « Beatcorp » telle que nous la pensons se base sur une architecture centralisée permettant à différents chercheurs d'accéder à distance à un ensemble de corpus et d'outils d'analyse partagés par d'autres chercheurs. En effet, la plateforme, à travers une application Web donne l'accès à la base de corpus, à l'ontologie, et à la base de scripts via le moteur de gestion. Notre objectif étant de permettre le partage entre chercheurs, l'utilisation naturelle de la plateforme devrait permettre à un chercheur d'accéder aux données déposées par un autre chercheur, aux scripts qu'il a définis, et aux descriptions des analyses réalisées. Cependant, un chercheur peut avoir une utilisation strictement locale et individuelle de la plateforme. Il peut aussi vouloir garder son travail privé pendant une période de test durant laquelle il peut faire des vérifications et apporter des modifications fréquentes ou parce qu'il pense que son travail n'est pas encore mature pour être partagé. Enfin, on peut imaginer qu'un chercheur dispose de données personnelles, voire sensibles qu'il ne souhaite pas partager du fait qu'elles peuvent porter atteinte à la vie privée des participants à une expérimentation. Le chercheur peut garder ces données privées et ne les partager qu'après anonymisation. La plateforme, pour être flexible, doit donc permettre à un chercheur d'avoir un espace privé dans lequel il stocke les données privées ou les données qu'il ne souhaite pas encore partager et qu'il peut analyser avec les outils d'analyse disponibles. Elle lui permet également d'accéder à l'espace public où il peut (1) consulter les données partagées par d'autres chercheurs auxquelles il a le droit d'accéder, et (2) déposer ses données pour les rendre accessibles à d'autres chercheurs autorisés.

8.7 Scénarios d'utilisation de la plateforme « Beatcorp »

Dans cette section, nous donnons trois exemples de scénarios d'utilisation de la plateforme « Beatcorp ». Le premier scénario consiste en la création d'un corpus initial relatif à une expérimentation d'apprentissage. Le deuxième scénario correspond à l'utilisation de la plateforme pour effectuer une requête de recherche dans la base de corpus. Enfin, le dernier

scénario concerne la création d'un corpus d'analyse de corpus existants partagés par d'autres chercheurs.

8.7.1 Création d'un corpus initial

Pour créer un corpus initial, le chercheur doit fournir, via une application Web, les métadonnées de description du corpus ainsi que les ressources qui le composent. Les figures (**Error! Reference source not found.** Figure 69 à Figure 76) ci-dessous illustrent par le biais d'une maquette d'interfaces Web les différentes étapes à suivre pour construire un nouveau corpus initial en suivant le modèle de corpus que nous proposons. Le chercheur commence donc par saisir les métadonnées de description générale du corpus (cf. Figure 69**Error! Reference source not found.**) (p. ex. titre, description, etc.). Pour certaines des données descriptives, telles que la description du corpus ou des objectifs d'apprentissage, le chercheur peut disposer de ressources contenant les données relatives. Il peut donc charger ces ressources de documentation au fur et à mesure de la description du corpus. La deuxième étape (cf. **Error! Reference source not found.**) permet au chercheur de sélectionner l'environnement d'apprentissage utilisé durant l'expérimentation relative au corpus, parmi les environnements d'apprentissage déjà décrits dans la plateforme. Si l'environnement d'apprentissage n'est pas encore décrit, le chercheur peut ajouter un environnement d'apprentissage et fournir les données de description correspondantes. La description de l'environnement d'apprentissage ne fait pas partie de la description du corpus. Au niveau de la description du corpus, l'environnement d'apprentissage utilisé sera référencé. La description de l'environnement d'apprentissage peut être utile lors de l'interrogation de la base de corpus.

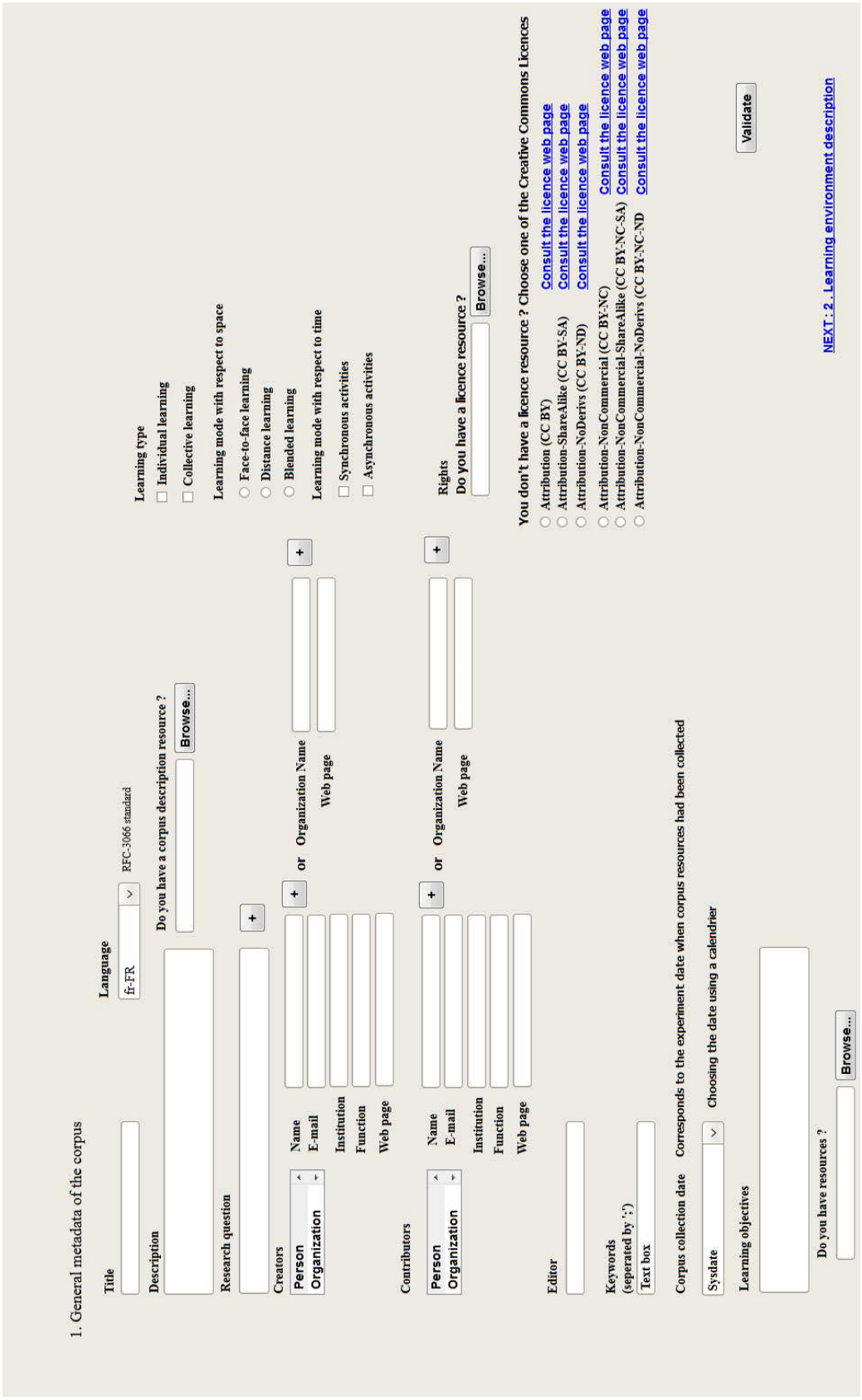


Figure 69 Maquette d'une interface Web de création d'un corpus initial (1/8)

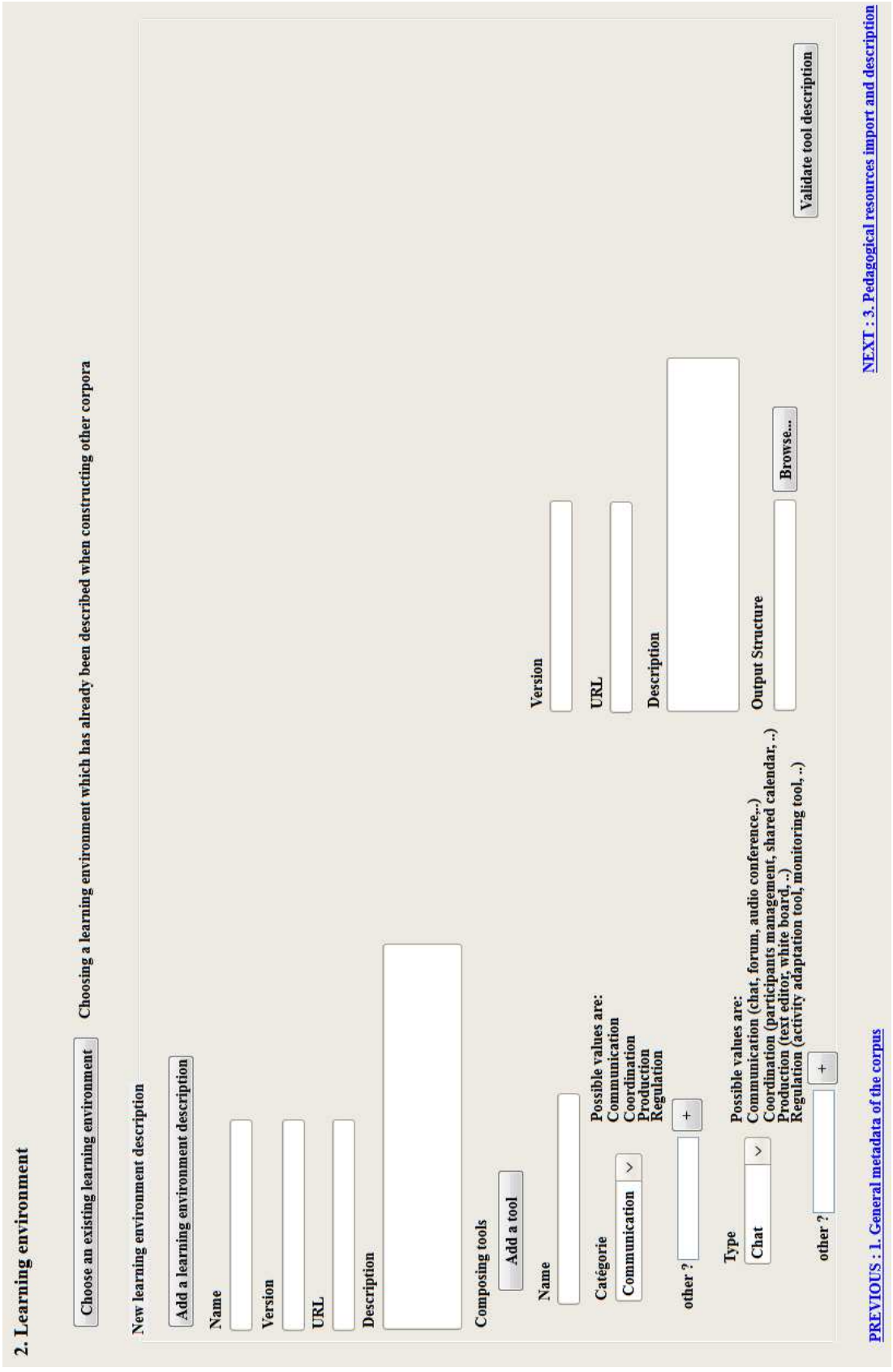


Figure 70 Maquette d'une interface Web de création d'un corpus initial (1/8)

Le chercheur doit ensuite importer et documenter les ressources pédagogiques offertes aux participants durant la session d'apprentissage (cf. Figure 71), les ressources de production résultant du travail des participants (cf. Figure 72), et les ressources traces collectées durant l'expérimentation d'apprentissage (cf. Figure 73). Enfin le chercheur peut décrire des analyses réalisées antérieurement à la création du corpus dans la plateforme. Pour ceci, il commence par sélectionner l'outil d'analyse utilisé parmi les outils d'analyse décrits dans la plateforme, ou si l'outil n'est pas encore décrit, il l'ajoute en fournissant les métadonnées nécessaires pour sa description en vue de partager ces données avec les autres chercheurs utilisant la plateforme (cf. Figure 74). Enfin, le chercheur décrit les analyses réalisées en fournissant les métadonnées nécessaires (p. ex. dates de début et de fin, créateurs, objectif, etc.), et en important les ressources utilisées et produites durant le travail d'analyse (cf. Figure 75 et Figure 76). Après la création du corpus, le chercheur étudie l'ontologie, et en particulier le modèle sémantique pour vérifier que les concepts qu'il souhaite interroger existent dans l'ontologie. Si ce n'est pas le cas, le chercheur peut enrichir l'ontologie. La dernière étape nécessaire pour permettre au chercheur d'interroger les ressources traces afin de les analyser, est de définir les scripts XQuery nécessaires pour l'extraction des données. Rappelons que pour être interrogées, les ressources traces doivent être structurées en XML. Ce choix étant technique, nous pouvons imaginer que des évolutions peuvent être apportées pour gérer d'autres types de représentations.

Figure 71 Maquette d'une interface Web de création d'un corpus initial (3/8)

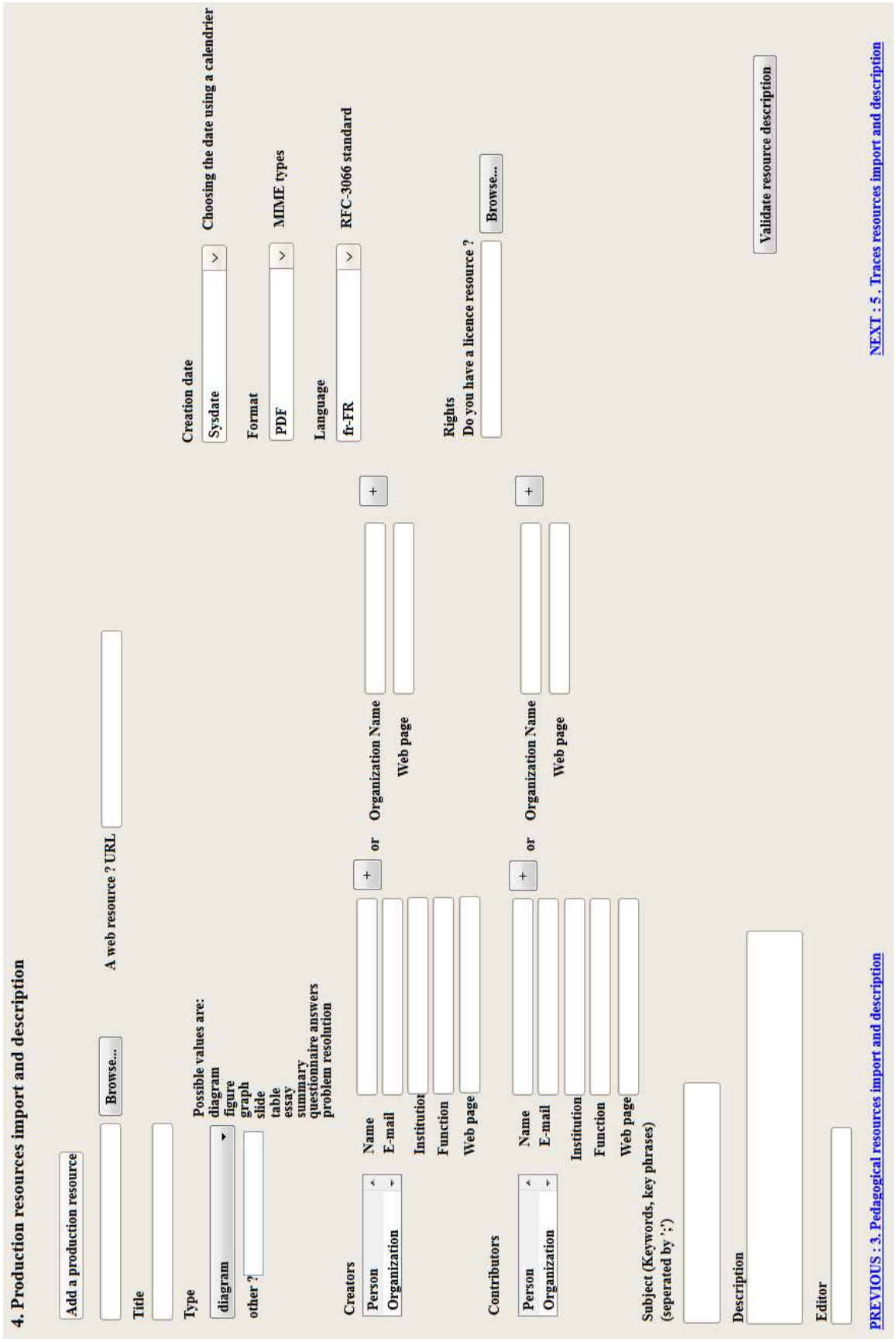


Figure 72 Maquette d'une interface Web de création d'un corpus initial (4/8)

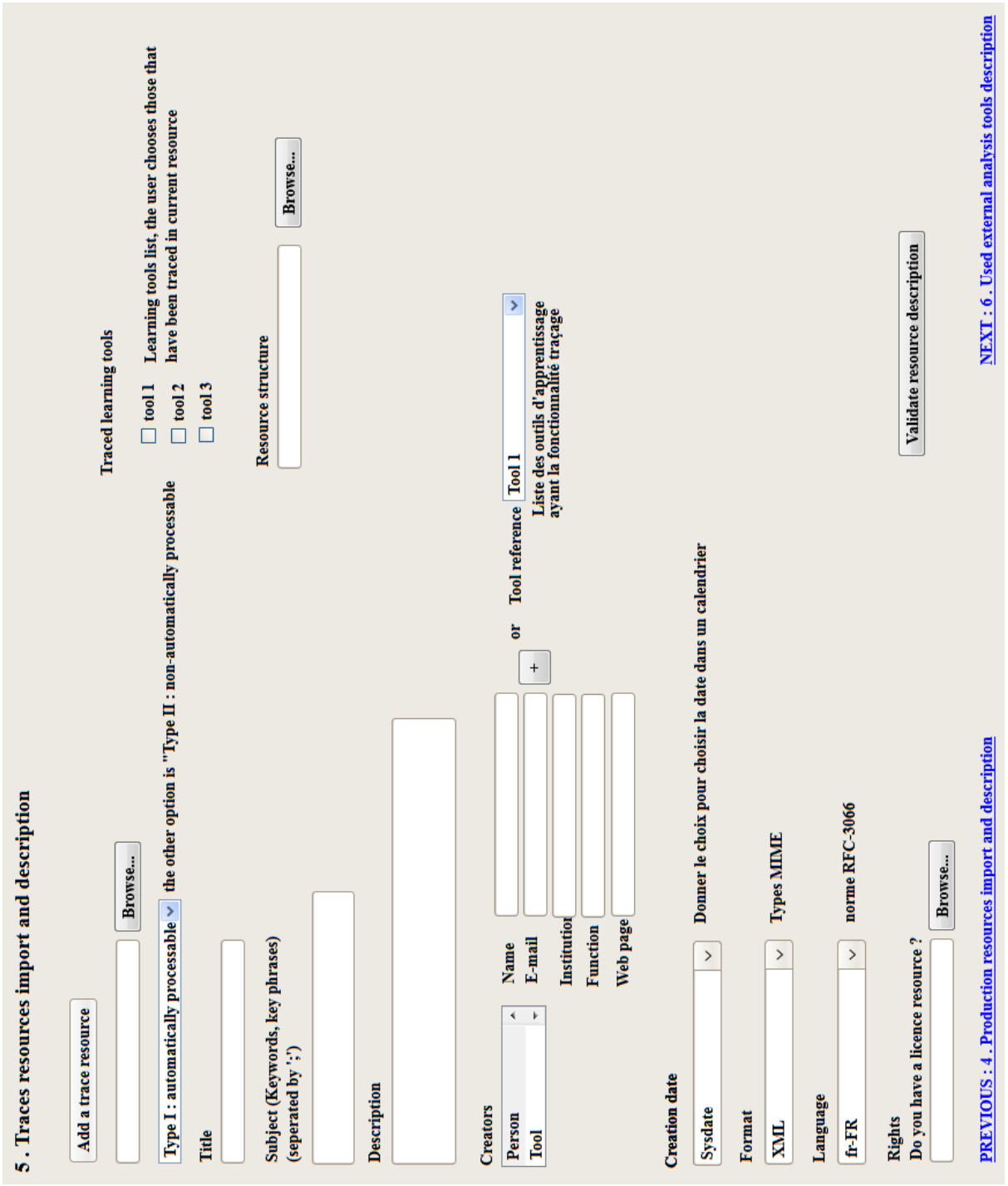


Figure 73 Maquette d'une interface Web de création d'un corpus initial (5/8)

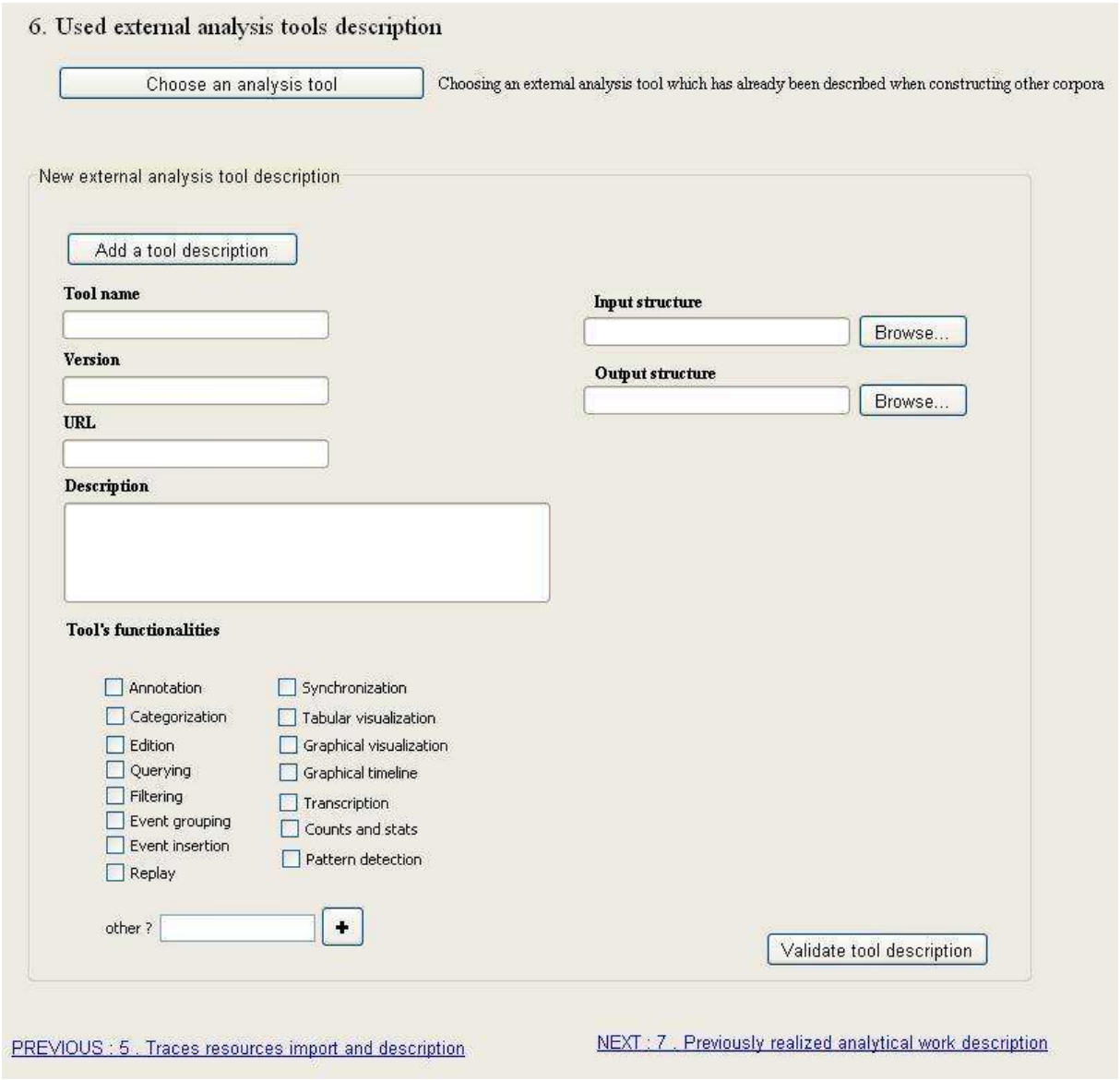


Figure 74 Maquette d'une interface Web de création d'un corpus initial (6/8)

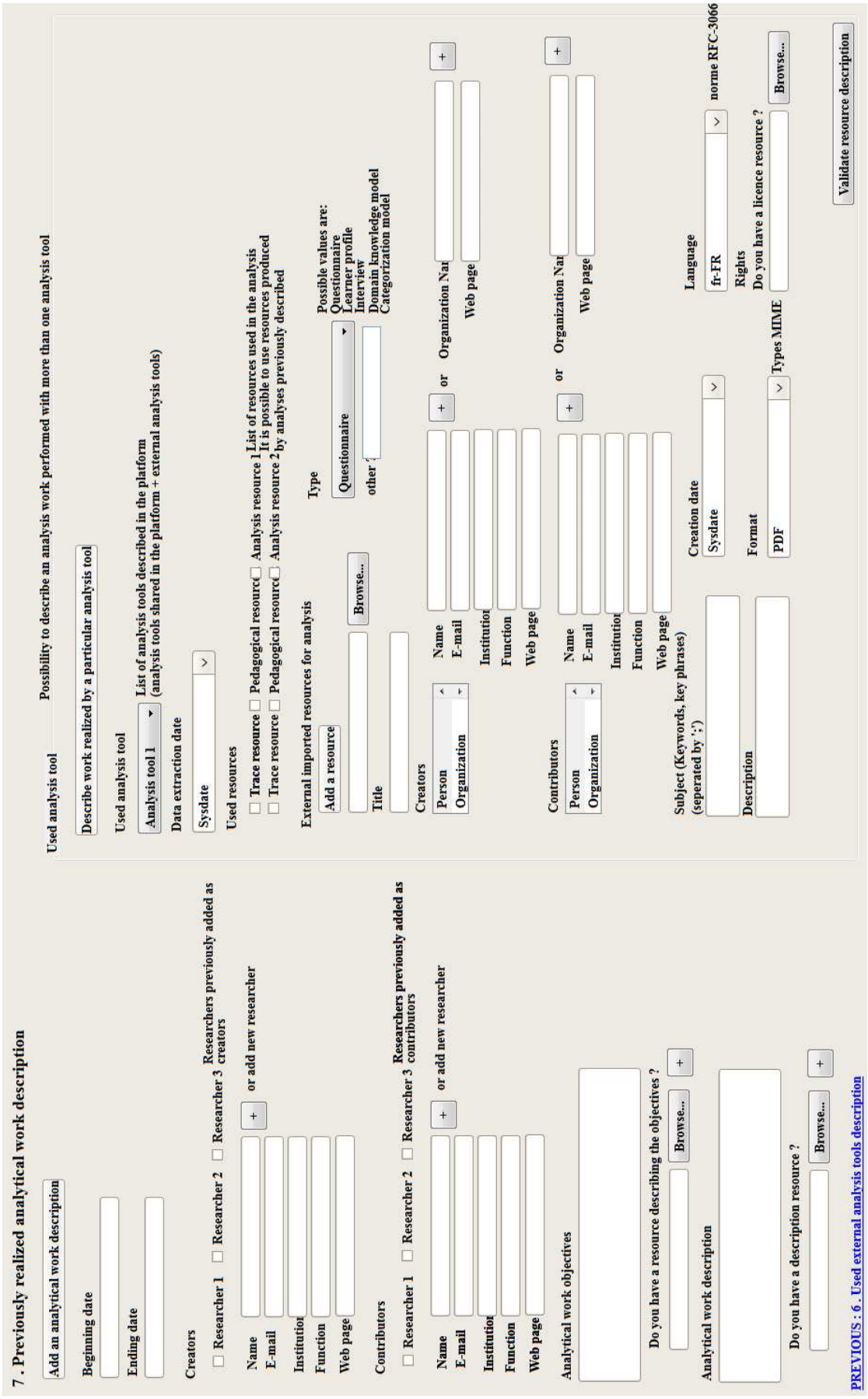


Figure 75 Maquette d'une interface Web de création d'un corpus initial (7/8)

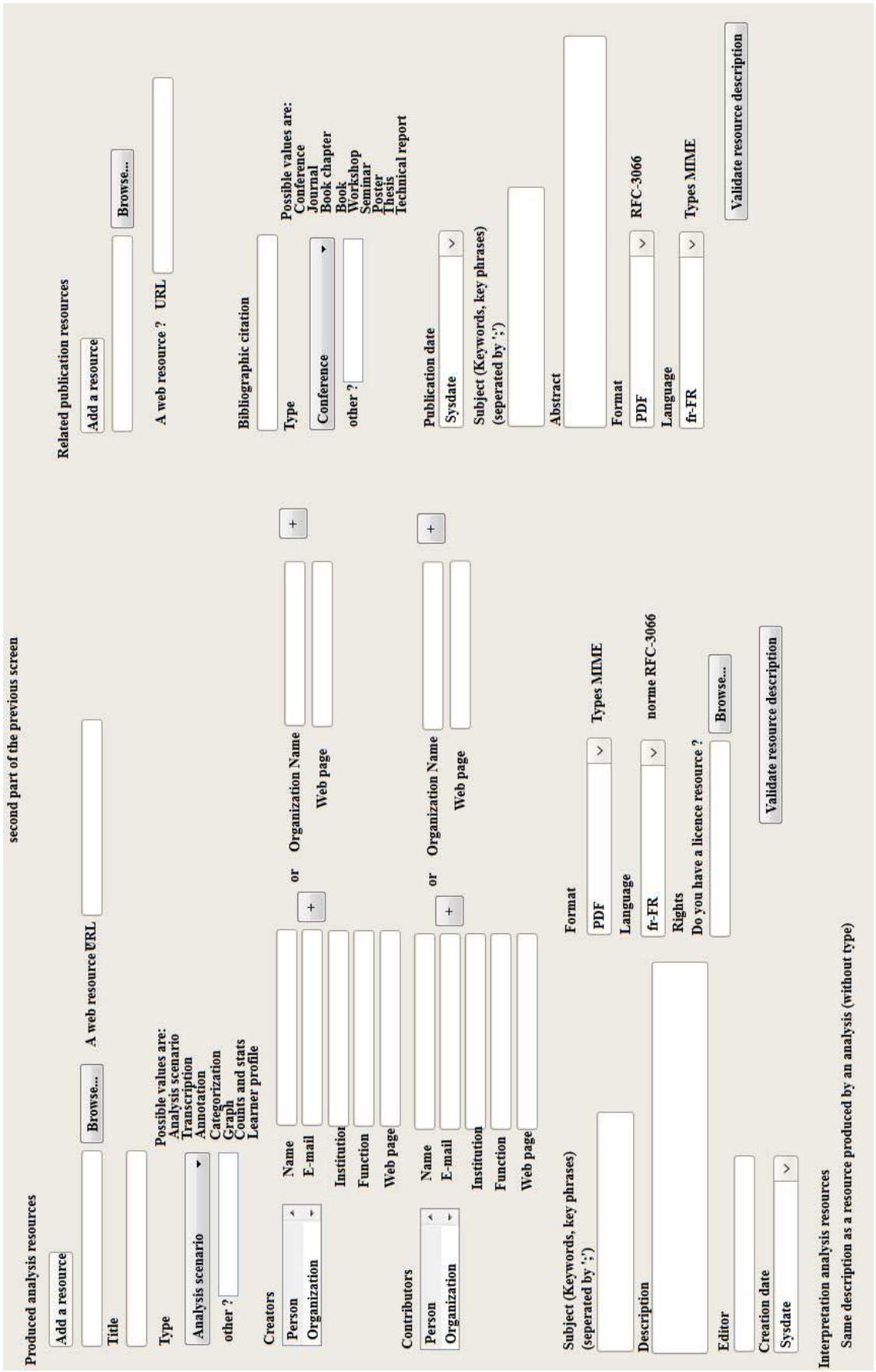


Figure 76 Maquette d'une interface Web de création d'un corpus initial (8/8)

8.7.2 Recherche dans la base de corpus

Pour effectuer une recherche dans la base de corpus, une interface basée sur le modèle de description de corpus proposé (cf. section 4.3.5) est offerte au chercheur. La Figure 77 ci-dessous illustre un exemple d'une telle interface. Le chercheur, choisit les critères de sa recherche. Plus les critères de recherche sont nombreux, plus la requête est restrictive. Les critères de recherche peuvent être liés aux métadonnées générales de description d'un corpus (p. ex. la date de création du corpus, les créateurs, et les contributeurs) ainsi qu'aux types des ressources partagées dans les corpus. Par ailleurs, un chercheur peut avoir besoin d'interroger la base de corpus en fonction des travaux d'analyse liés aux corpus (fonctionnalités des outils d'analyse utilisés, types des ressources d'analyse, et types des publications liées au travail d'analyse). Des critères spécifiques aux corpus initiaux peuvent également être utilisés dans la recherche, tels que la date de collecte des corpus, le type d'apprentissage, le mode d'apprentissage, les types de ressources partagées d'apprentissage et de production, et les types des outils composant les environnements d'apprentissage. Enfin, un chercheur peut effectuer une recherche textuelle permettant de rechercher des mots clés dans les métadonnées de description des corpus.

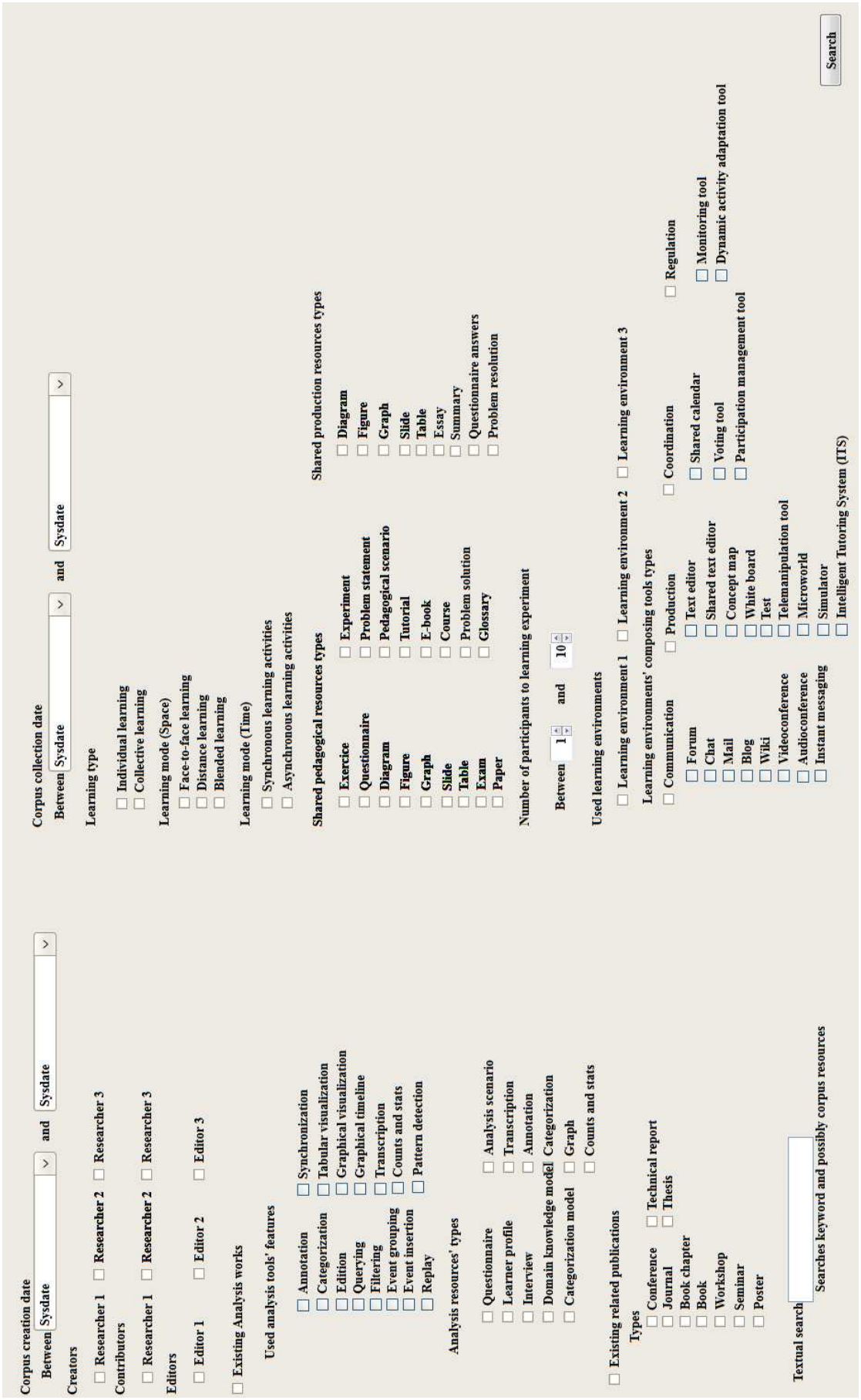


Figure 77 Maquette d'une interface Web d'interrogation de la base de corpus

8.7.3 Création d'un corpus d'analyse et analyse de corpus existants

La création d'un corpus d'analyse est liée à une nouvelle question de recherche étudiée. La Figure 78 ci-dessous illustre l'interface de création d'un corpus d'analyse. Elle permet au chercheur de fournir les métadonnées de description du corpus telles que le titre, la question de recherche étudiée, la description du corpus, les créateurs, les mots-clés, et les droits sur le corpus. Un corpus d'analyse pouvant contenir plusieurs travaux de recherche liés à la question de recherche étudiée, la Figure 79 ci-dessous illustre l'interface permettant au chercheur de lancer un nouveau travail d'analyse. Le chercheur sélectionne le corpus d'analyse qui va contenir son travail d'analyse. Il fournit ensuite les métadonnées de description du travail d'analyse telles que les contributeurs, les objectifs d'analyse, et les dates de début et de fin. L'étape suivante consiste à choisir l'outil d'analyse à utiliser parmi ceux qui sont partagés par la plateforme, et les corpus dont les ressources vont être analysées. Pour pouvoir utiliser un outil dans l'analyse d'un ou plusieurs corpus, les scripts d'extraction, de filtrage et de formatage nécessaires doivent être définis précédemment pour permettre de préparer les données à analyser. Le chercheur, après avoir sélectionné l'outil d'analyse à utiliser, identifie les scripts nécessaires pour la préparation des données. Il peut importer des ressources externes supplémentaires utiles pour son travail. Enfin, si l'outil utilisé permet de sauvegarder le résultat du travail réalisé, le chercheur importe les ressources produites durant l'analyse. Les éventuelles ressources d'interprétation et publications liées peuvent aussi être importées et décrites pour enrichir le corpus et faciliter la compréhension, la comparaison et la reproductibilité des résultats. La sélection des scripts (cf. Figure 79) utilisés dans l'extraction, la conversion, et le formatage des données à analyser diffèrent suivant le type d'outil d'analyse utilisé. Par exemple, si l'outil d'analyse est générique (p. ex. Tatiana (Dyke et al. 2009)) et peut être utilisé dans l'analyse de différents types de données, il revient à l'utilisateur d'identifier quels concepts correspondent aux données qui l'intéressent. Cependant, beaucoup d'outils d'analyse sont spécifiques à des types particuliers de traces (p. ex. les outils de la plateforme CALICO (CALICO, 2010) qui s'intéressent aux communications de forums) et leur utilisation sur un corpus particulier peut être automatique (à partir du moment où les scripts nécessaires sont définis) sans demander à l'utilisateur de sélectionner les scripts à utiliser. Même dans le premier cas, la sélection des scripts à utiliser peut devenir transparente et être remplacée par un outil permettant au chercheur de

sélectionner les concepts du modèle sémantique à interroger. C'est le proxy de l'outil d'analyse qui se chargera par la suite de composer les scripts à utiliser.

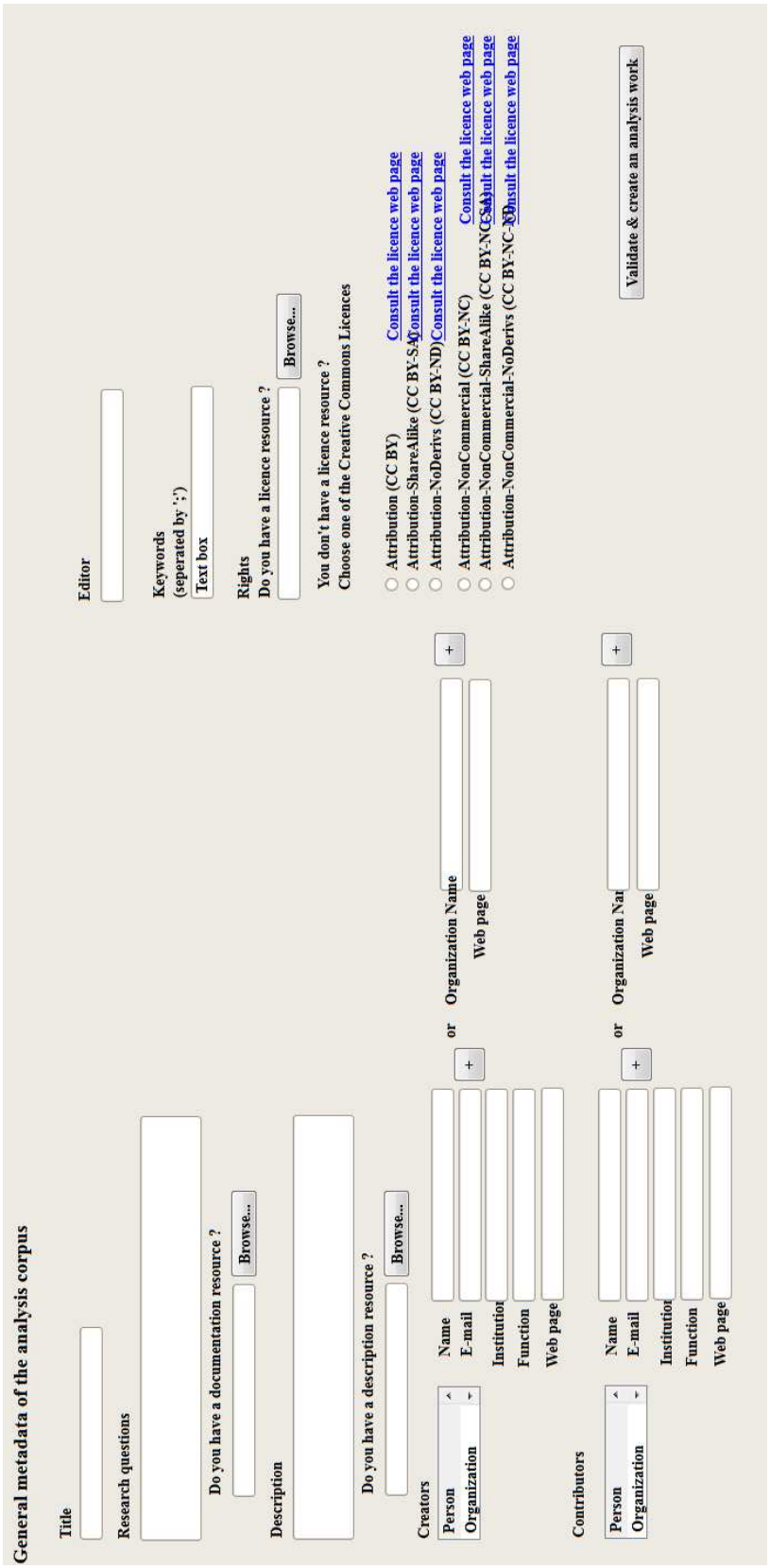


Figure 78 Maquette d'une interface Web de création d'un corpus d'analyse

Figure 79 Maquette d'une interface Web pour réaliser un nouveau travail d'analyse

8.7.4 Ajout d'un nouvel outil d'analyse

L'un des objectifs de notre travail étant de permettre le partage et la réutilisation d'outils d'analyse entre chercheurs, nous considérons le scénario d'ajout d'un outil d'analyse à la plateforme. La Figure 80 ci-dessous illustre une interface d'ajout d'un nouvel outil d'analyse. Le chercheur doit donc renseigner quelques métadonnées descriptives de l'outil et de ses fonctionnalités et importer d'éventuelles ressources de documentation nécessaire à l'utilisation de l'outil. Si une structure formalisée de l'entrée de l'outil d'analyse existe, celle-ci doit être fournie car elle sera indispensable à l'étape de création des scripts du « proxy » permettant d'interface la base des corpus et l'outil d'analyse ajoutée. L'ajout d'un outil d'analyse est suivi d'un travail d'alignement sémantique entre la structure des données d'entrée de l'outil et les concepts du modèle sémantique correspondant. Il est possible que des concepts non existants soient définis à la demande du chercheur. L'étape suivante consiste à vérifier si les scripts d'interrogation des concepts retenus pour analyser un ou plusieurs corpus sont définis pour ceux-ci, auxquels cas, ils pourraient être réutilisés. Sinon une définition des scripts nécessaires doit être réalisée par le chercheur. Les scripts d'extraction, de filtrage, et de formatage spécifiques à l'outil d'analyse doivent également être définis pour compléter les fonctionnalités du « proxy » permettant un interfaçage complet corpus/outil d'analyse.

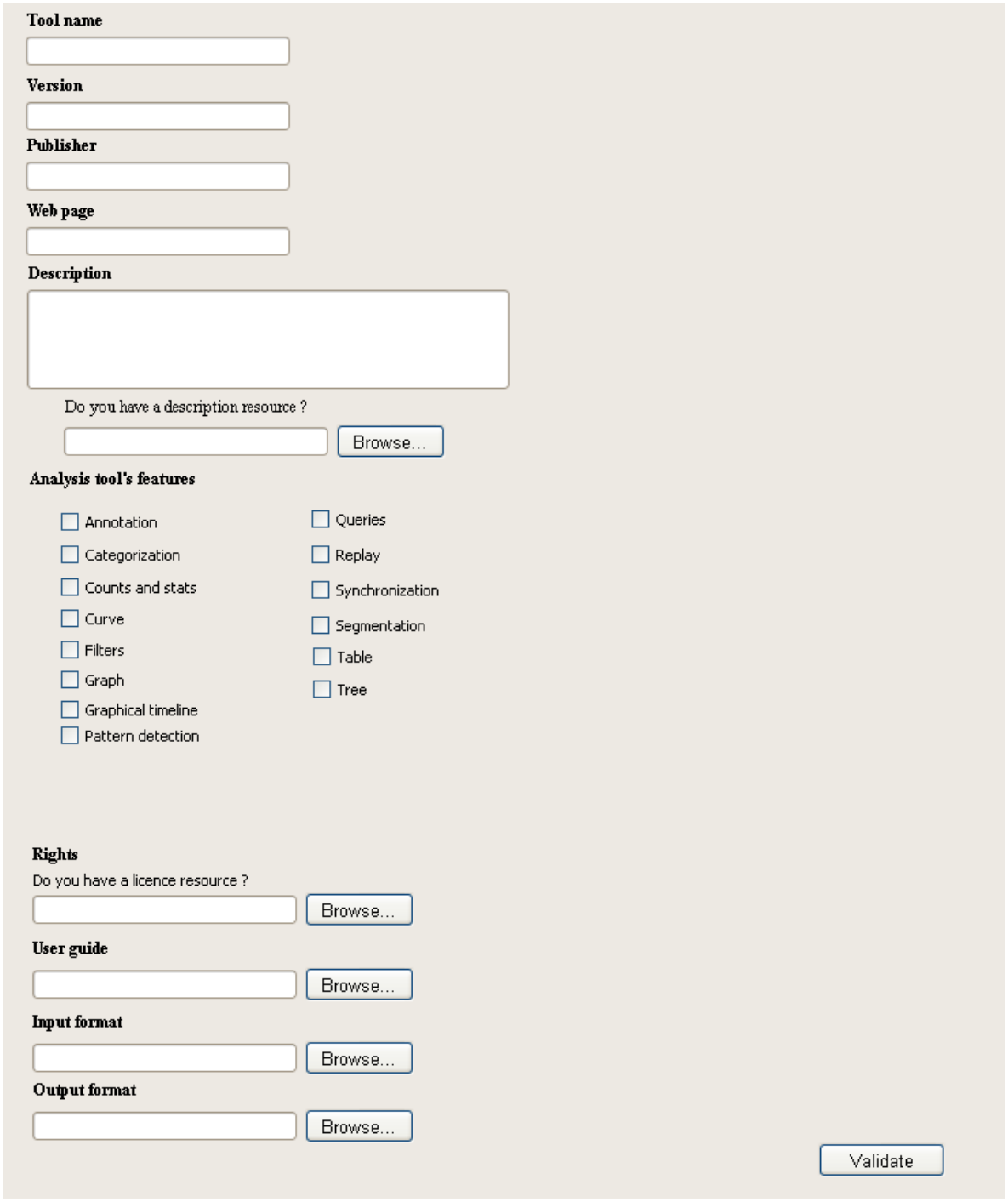


Figure 80 Maquette d'une interface Web pour ajouter un nouvel outil d'analyse

8.8 Positionnement par rapport à l'existant

Nous reprenons maintenant les différents points clefs évoqués dans notre analyse du chapitre 2, et tentons de voir comment l'approche « Proxyma » et la plateforme « Beatcorp », qui se base sur cette approche, peuvent satisfaire nos critères. Dans la Figure 81 ci-après, nous complétons le tableau présenté dans le paragraphe 2.3.9 du chapitre 2 en intégrant notre

approche pour l'évaluer suivant les critères proposés. Notre approche permet naturellement de gérer des traces hétérogènes tout en préservant la richesse des traces initiales en évitant de les convertir dans une représentation particulière. En effet, du fait que nous n'imposons pas de représentation standard, différentes traces hétérogènes provenant de différents systèmes et relatives à différents domaines d'application peuvent être prises en compte. Concernant le troisième critère « extensibilité du format de représentation des traces », nous considérons qu'on y répond malgré le fait qu'on ne définit pas de représentation spécifique pour les traces. En effet, ce critère exprime le fait que des traces ayant des modèles différents de celui proposé par un projet puissent être prises en charge. Dans notre cas, nous n'imposons pas de représentation des traces, mais définissons, au moyen d'une ontologie, un ensemble de concepts pouvant être retrouvés dans ces traces. La définition de ces concepts fait partie d'un processus incrémental et participatif consistant à définir les concepts au fur et à mesure que le besoin est exprimé. Concernant la contextualisation des données partagées dans les corpus, le modèle de corpus que nous proposons tente de collecter un ensemble de données contextuelles structurées à travers les métadonnées utilisées dans la description d'un corpus, des ressources qui le composent et des analyses réalisées. Par ailleurs, parmi les types de ressources définis par ce modèle on trouve les ressources de documentation pouvant contenir des descriptions d'une expérimentation et des analyses (par exemple, une communication scientifique). De plus, un corpus peut contenir des ressources relatives à la situation d'apprentissage telles que des ressources pédagogiques offertes par l'environnement d'apprentissage pour assurer le bon déroulement de ce dernier, ainsi que les ressources produites par les participants à l'apprentissage au cours de leur activité. Notre approche permet de gérer des corpus relatifs à des sessions d'apprentissage individuel ou collectif. Dans le cadre de la validation de ce travail, les corpus que nous avons construits correspondent à des situations d'apprentissage collectives. Mais l'approche peut très bien s'appliquer à des situations individuelles. Il suffirait de vérifier que les concepts nécessaires à l'interrogation des traces contenues dans le corpus sont définis dans le modèle sémantique et que les scripts nécessaires à l'extraction des données correspondantes sont définis dans le modèle opérationnel, ou de les définir si ce n'est pas le cas. Notre objectif étant de permettre l'interfaçage tant que possible entre les corpus et les outils d'analyse, notre approche permet de gérer des outils d'analyse générique pouvant s'appliquer à des traces provenant d'environnements d'apprentissage variés ou des outils d'analyse plus spécifiques liés à un domaine d'application particulier. Le partage des corpus entre les chercheurs devient de plus en plus intéressant lorsque les analyses réalisées sur ces corpus sont partagées et décrites. Ce

qui permet de comparer, enrichir, et reproduire des analyses existantes. Le modèle de description de corpus que nous proposons permet de lier et décrire les analyses réalisées sur les corpus tout en faisant référence aux scripts utilisés dans l'extraction des données ce qui permet de reproduire ces analyses de manière automatique et d'expliciter les données réellement utilisées dans l'analyse.

Critères / Projet	Hétérogénéité des traces	Préservation de la richesse des traces initiales collectées	Extensibilité du format de représentation des traces	Contextualisation des traces	Type apprentissage		Outil d'analyse		Partage des analyses	Reproductibilité des analyses
					Individuel	collectif	Générique	Spécifique		
IA JEIRP	+	-	+	-	-	+	-	+	-	+
PSLC Datashop	+	-	+	+	+	-	-	+	+	+
REDiM	+	+	-	-	+	+	-	-	-	-
CALICO	-	-	-	-	-	+	+	-	-	+
MULCE	+	-	+	++	-	+	-	-	+	+
Undertracks	+	-	+	+	+	-	+	+	+	+
dataTel	-	-	-	+	+	-	+	-	-	-
CAM-CIM	+	-	+	-	+	-	-	-	-	-
Proxyma	+	+	+	+	+	+	+	+	+	+

Figure 81 Tableau récapitulatif de l'évaluation des travaux étudiés et qui traitent du partage de corpus et de leur analyse, comparaison avec Proxyma

8.9 Conclusion

Ce chapitre présente une proposition d'architecture de la plateforme « Beatcorp » de partage et d'analyse de corpus de traces d'interaction contextualisées. Cette architecture se base sur les trois modèles de l'approche « Proxyma » proposée dans ce travail. L'architecture

est composée de cinq composants : une base de corpus pour le stockage des corpus, une ontologie pour la formalisation des modèles, une base de scripts pour le stockage des scripts permettant l'exploitation des corpus, un moteur de gestion permettant la manipulation des différents composants de la plateforme, et une application Web jouant le rôle d'interface entre le chercheur et les différents composants de la plateforme lui permettant de construire des corpus, de parcourir les corpus partagés, et de les interroger. Les scénarios d'utilisation décrits illustrent trois fonctionnalités essentielles offertes par la plateforme « Beatcorp ». Nous avons démontré à la fin de ce chapitre que l'approche « Proxyma » répond aux différents critères que nous avons identifiés pour répondre à nos questions de recherche.

Chapitre 9 : Exemples d'application de l'approche

9.2 Introduction

Le montage d'une expérimentation d'apprentissage écologique, c'est-à-dire dans un vrai contexte d'apprentissage, peut être très coûteux en termes de temps de par la complexité de l'organisation nécessaire. En effet, un tel travail nécessite généralement la mobilisation d'une équipe composée de plusieurs personnes. Par exemple, un ou plusieurs chercheurs et éventuellement des enseignants se chargent, en fonction des questions de recherche étudiées, de la formalisation des besoins de l'expérimentation ainsi que des scénarios d'apprentissage à suivre, un ou plusieurs développeurs mettent en place l'environnement d'apprentissage utilisé, un ou plusieurs techniciens pour l'installation et le paramétrage de l'environnement ainsi que la création des comptes utilisateurs, et les participants/apprenants à l'expérimentation

d'apprentissage, cette dernière pouvant concerner un module de leur cursus scolaire. Une fois l'expérimentation finie, un travail de collecte et de construction de corpus reste à faire par les chercheurs.

Dans le cadre de ce travail de thèse, pour valider notre approche, nous construisons deux corpus initiaux et un corpus d'analyse. Pour construire le premier corpus initial, intitulée « EMSE-LEAD » nous avons collecté les ressources correspondant à une expérimentation réalisée en 2006 en utilisant l'environnement d'apprentissage DREW (Corbel et al., 2003) dans le cadre du projet LEAD mené par d'autres chercheurs d'une équipe participant au projet « personnalisation des EIAH »[12] ayant comme objectif d'analyser l'interaction entre participants à une session d'apprentissage collaboratif en face à face. Le deuxième corpus initial, intitulé « COO-POO », est relatif à une expérimentation que nous avons menée en utilisant la plateforme d'apprentissage Moodle (Moodle, 2013) dans le cadre du module « Conception orientée objet, Programmation orientée objet » avec un groupe d'étudiants en multimédia à l'IUT de Chambéry. Enfin, le corpus d'analyse que nous avons construit a pour but d'analyser les traces collectées dans le corpus « COO-POO » pour évaluer le rôle du forum dans l'amélioration de la collaboration apprenant/apprenant et apprenant/enseignant. Ces trois corpus sont présentés dans la suite de ce chapitre.

9.3 Construction de corpus

Le modèle de corpus que nous proposons dans le chapitre 4 définit un modèle de description d'un corpus et des ressources qui y sont partagées. Ce modèle offre la possibilité aux chercheurs de documenter leurs corpus en important des ressources existantes et/ou en saisissant des métadonnées textuelles lors de la construction des corpus. Pour définir ce modèle, nous avons identifié les types de ressources pouvant exister dans un corpus de traces d'interaction et des métadonnées utiles pour la description d'un corpus et des ressources qui le composent. Un chercheur souhaitant partager son corpus fournit des métadonnées générales sur le corpus, importe les ressources qui composent son corpus et les décrit par des métadonnées (cf. 4.3.5.2).

Des chercheurs qui collectent et exploitent un corpus de traces d'interaction d'apprentissage dans le cadre d'un travail de recherche n'ont pas forcément l'intention de le

[12] http://cluster-isle-eiah.liris.cnrs.fr/

partager. De plus, vu la complexité du travail d'organisation et de documentation d'un corpus partagé qui nécessite beaucoup de temps, les données restent souvent éparpillées et risquent même de devenir obsolètes après quelques années (par exemple, suite au départ en retraite du chercheur qui en est à l'origine ou à l'évolution des thématiques de recherche du laboratoire). La documentation d'un corpus en vue de le partager avec d'autres chercheurs est donc la plupart du temps absente et se limite à une description de l'expérimentation sous forme d'un article scientifique, souvent très concise à cause du nombre de pages limité. La reconstruction d'un corpus peut être l'occasion pour le chercheur de rassembler toutes les ressources relatives à une expérimentation, de les organiser et de les décrire afin qu'elles soient exploitables par d'autres chercheurs.

Nous réutilisons la plateforme eXist-db (eXist-db, 2012) de gestion de base de données XML pour stocker les corpus que nous construisons. Dans un eXist-db, les documents (XML ou autres) stockés dans la base de données, sont organisés dans des collections. C'est une organisation semblable aux systèmes de fichiers des systèmes d'exploitation. Une collection peut contenir des fichiers de données (XML ou autres p. ex. un document PDF, un enregistrement audio/vidéo, etc.) mais aussi d'autres collections. Un corpus correspond donc à une collection dans la base de données eXist-db, qui contient à son tour d'autres collections permettant d'organiser les corpus et les ressources qui les composent.

9.4 Corpus EMSE-LEAD – Construction, interrogation, et analyse

Nous présentons dans cette section le premier exemple de corpus initial généré par l'environnement d'apprentissage collaboratif DREW. Nous commençons par la présentation de DREW et de l'expérimentation d'apprentissage à l'origine du corpus. Nous décrivons ensuite la composition du corpus établie suivant le modèle de corpus (cf. chapitre 4). Enfin, nous donnons quelques exemples de scripts permettant d'interroger les traces partagées dans le corpus pour extraire les interactions relatives au *chat*.

9.4.1 Présentation de l'environnement DREW

DREW (Dialogical Reasoning Educational Web tool ou outil Web pour l'apprentissage du raisonnement dialogué) (Corbel et al., 2003) est un environnement informatique

d'apprentissage humain collaboratif développé dans le cadre du projet européen SCALE[13] (Support Collaborative Argumentation-Based Learning ou support de l'apprentissage collaboratif basé sur l'argumentation). L'environnement DREW est composé de deux types d'interface. La première interface est destinée aux apprenants et offre les outils de communication et de production collaborative suivants : un outil de *chat*, un tableau blanc, un éditeur de texte individuel ou collectif et un éditeur de graphe d'argumentation. La deuxième interface est destinée aux enseignants/chercheurs et leur permet la création de tâches d'apprentissage et le choix des outils de communication et de production collaborative à utiliser par les apprenants. Elle permet également le rejouage de sessions précédentes à partir des fichiers de traces qui enregistrent les interactions passées.

9.4.2 Présentation de l'expérimentation

L'expérimentation considérée utilise l'environnement d'apprentissage DREW dans le cadre d'une activité d'encadrement en face à face d'un ensemble de diades pour la réalisation de projets de programmation en langage C par des élèves ingénieurs de l'Ecole des Mines de Saint-Etienne. Les étudiants étaient appelés à communiquer en utilisant les outils offert dans l'environnement DREW, en particulier un outil de chat et un éditeur de texte partagé. Cette expérimentation a été réalisée dans le cadre du projet LEAD[14] (Technology-enhanced learning and problem-solving discussions) en collaboration entre les laboratoires RIM de l'Ecole des Mines de Saint-Etienne et ICAR de l'Université Lyon 2. Cette expérimentation a été montée dans le cadre du travail de thèse de Gregory Dyke (Dyke, 2009) (Dyke et al., 2009), et le corpus résultant a été utilisé pour tester les fonctionnalités de l'outil TATIANA d'analyse des interactions développé dans le cadre de son travail. L'outil de communication « chat » et l'outil de production collaborative « éditeur de texte partagé » de DREW sont mis à la disposition des apprenants de chaque diade. Le chat permet aux apprenants de discuter, tandis que l'éditeur de texte partagé est utilisé pour prendre des notes sur les explications et les consignes données par l'encadrant. Ce dernier pouvait aussi intervenir dans le chat. Les participants pouvaient communiquer verbalement, mais les apprenants étaient encouragés à prendre des notes dans l'éditeur de texte partagé.

[13] http://scale.emse.fr/
[14] http://lead.emse.fr/

9.4.3 Le corpus

Ce corpus est constitué des ressources provenant de l'expérimentation décrite dans le paragraphe précédent. Nous n'avons pas trouvé de ressources de documentation à part une description concise de l'expérimentation dans des publications scientifiques (thèse de Gregory Dyke (Dyke, 2009), article (Dyke et al., 2009)). Nous avons récupéré les ressources qui étaient disponibles et qui consistent en une vidéo qui enregistre une séance d'encadrement ainsi qu'un fichier généré par l'environnement DREW et qui enregistre les traces d'utilisation du chat et de l'éditeur de texte relatives à la même session d'encadrement. Nous avons également récupéré une ressource de traces DREW relatives à une autre session d'encadrement. Cette dernière n'est pas accompagnée d'un enregistrement vidéo. Par ailleurs, nous avons récupéré trois ressources relatives à des analyses réalisées sur les ressources de traces collectées. Ces ressources ont été créées dans l'outil d'analyse Tatiana (Dyke et al., 2009) et peuvent donc être importées dans l'outil pour être consultées ou enrichies par des analyses complémentaires. La première ressource représente la transcription de l'enregistrement vidéo en utilisant l'outil ELAN[15] d'annotation de ressources audio/vidéo. La deuxième ressource d'analyse représente une catégorisation des notes prises par les apprenants durant la session d'encadrement. Et la troisième ressource représente un graphe de reformulation illustrant le lien entre les notes prises par les apprenants et la transcription du discours de l'enseignant encadrant. Après avoir récupéré ces ressources, nous les avons stockées dans une collection de la base de données eXist. La base de données eXist permet d'organiser des fichiers dans des collections, une collection pouvant être assimilée à un dossier. Nous définissons une collection « corpus initiaux » pour contenir les corpus initiaux du composant « base de corpus ». Cette collection est composée de sous-collections, chacune étant relative à un corpus initial particulier. Pour construire le corpus de l'expérimentation réalisée à l'aide de l'environnement d'apprentissage DREW, nous créons une collection que nous appelons « EMSE-LEAD » désignant le corpus initial relatif à l'expérimentation et qui est une sous-collection de la collection « corpus initiaux ». Cette collection (cf. Figure 82) contient un fichier RDF décrivant le corpus, les ressources qu'il contient, et les analyses réalisées sur le corpus. Elle contient également des sous collections contenant les différents types de ressources physiques collectées. Nous avons donc créé trois sous-collections. La première « ressources traces », elle-même composée de deux sous-collections, (1) la première « ressources traces automatiquement traitables » contenant les deux fichiers de type log

[15] http://icar.univ-lyon2.fr/projets/corinte/confection/elan.htm

collectés par l'environnement DREW et pouvant être traitée automatiquement par un programme sans avoir besoin d'un traitement manuel préalable, et (2) la deuxième « ressources traces non automatiquement traitables » contenant le fichier d'enregistrement vidéo d'une séance d'encadrement devant être transcrite avant tout traitement automatique. La deuxième sous-collection « ressources d'analyse » contient les trois ressources d'analyse collectées. Et la troisième sous-collection « ressources publication » contient les deux publications récupérées.

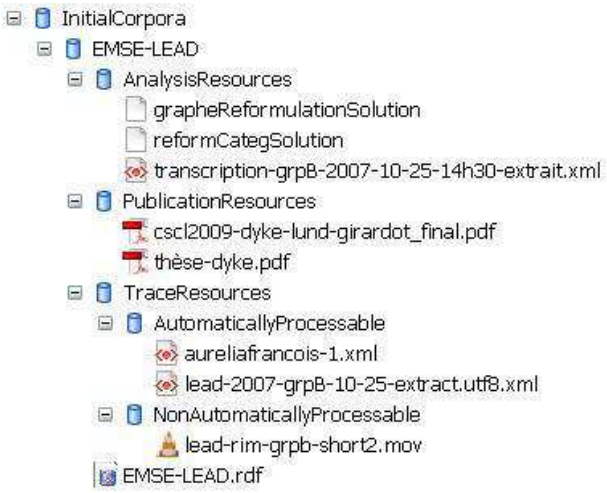


Figure 82 Composants du corpus « EMSE-LEAD »

9.4.4 Interrogation du corpus

L'interrogation d'un corpus repose sur le modèle sémantique (cf. Chapitre 5) et le modèle opérationnel (cf. Chapitre 6) définis dans l'approche « Proxyma ». Le modèle sémantique définit des concepts partagés entre chercheurs pour interroger les corpus. Le modèle opérationnel quant à lui définit six types d'opérations permettant de définir des scripts assurant un alignement entre le contenu d'un corpus, les concepts définis par le modèle sémantique, et les données attendues à l'entrée d'un outil d'analyse. Nous avons défini un ensemble de scripts écrits en XQuery permettant d'interroger le corpus pour extraire les interactions relatives aux discussions tracées dans l'outil de *chat* de l'environnement DREW.

DREW

Simple concept querying script corresponding to MessageContent concept

```
declare function drew:chatMessageContent($doc as xs:string) as xs:string*
{
    let $msg := doc($doc)//chat/text
    for $i in $msg
    return
    if (empty($i/text()))
    then ""
    else
    $i/text()
};
```

Complex concept querying script corresponding to ChatInteraction concept

```
declare function drew:chatInteraction($doc as xs:string) as node()*
{
    let $chatMessagesSenders := drew:chatMessageSender($doc), $chatTemporalIndicators :=
    drew:chatTemporalIndicator($doc),$chatMessages := drew:chatMessage($doc)
    for $i at $c in $chatMessages
    return
    <ChatInteraction>
    {
      $chatMessagesSenders[$c],
      $chatTemporalIndicator[$c],
      $chatMessages[$c]
    }
    </ChatInteraction>
};
```

Exrtracting script retrieving all chat interactions from a list of resources, calls ChatInteraction querying script (no need for data type conversion)

```
declare function drew:chatInteractions($docs as xs:string*) as node()*
{
    for $i in $docs
    return
    drew:chatInteraction($i)
};
```

Filtering script retrieving all chat interactions corresponding to sending messages, executed on the output of chatInteractions extracting script

```
declare function drew:chatMessageSendingInteractions( $chatInteractions as node()* ) as
node()*
{
        for $i at $j in $chatInteractions
        return
        if(empty($i/ChatMessage/MessageContent/text()))
        then ()
        else $chatInteractions[$j]
};
```

Figure 83 Exemples de (1) scripts d'interrogation de concept simple et complexe, (2) script d'extraction, et (3) script de filtrage, relatifs aux traces d'interactions de chat de DREW

La Figure 83 ci-dessus illustre des exemples de scripts permettant l'interrogation des concepts relatifs aux interactions de chat, l'extraction des données relatives aux interactions de chat à partir d'un ensemble de ressources et le filtrage de ces données extraites pour ne garder que les interactions relatives à l'envoi de message (en effet, les traces de chat contiennent souvent les événements relatifs à la connexion/déconnexion). Le premier script est un exemple de script d'interrogation du concept simple « contenu de message » qui

représente le message textuel envoyé dans un chat. Ce concept est un concept simple constituant le concept complexe « message de chat » avec d'autres concepts simples tels que le concept « chatroom » permettant d'identifier le salon de clavardage dans lequel un message a été envoyé, et le concept « identifiant du message » permettant de donner un identifiant unique à un message envoyé. La Figure 84 illustre un extrait du modèle sémantique relatif à la définition du concept « message de chat ». Le deuxième script est un exemple de script d'interrogation du concept complexe « interaction dans le chat » composé, entre autres, des concepts « message de chat », « expéditeur du message » et « indicateur temporel ». La Figure 85 illustre un extrait du modèle sémantique relatif à la définition du concept « interaction dans le chat ». Le troisième script est un exemple de script d'extraction permettant de renvoyer à partir d'un ensemble de ressources les données relatives aux interactions de chat. Le quatrième script est un exemple de script de filtrage permettant de retourner uniquement les interactions relatives à l'envoi de message dans le chat. Une interaction dans le chat pouvant être relative à un événement de connexion ou déconnexion, il s'agit de vérifier que le message relatif à l'interaction n'est pas vide. Tous les scripts données en exemple sont liés à la structure des traces collectées par l'environnement d'apprentissage DREW. Cette structure est définie sous forme d'un document DTD.

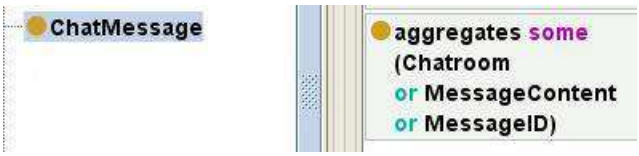


Figure 84 Extrait du modèle sémantique : définition du concept complexe « message de chat », concept constitué des concepts simples « chatroom », « contenu message », et « id message »

Figure 85 Extrait du modèle sémantique : définition du concept complexe « interaction de chat » défini comme l'agrégation de concepts complexes et de concepts simples

Dans ce premier exemple de corpus initial, des analyses préalables à la construction du corpus dans la plateforme « Beatcorp » sont déjà réalisées. Il est donc intéressant de décrire ces analyses et de partager les ressources qui y sont relatives pour que d'autres chercheurs puissent les consulter et les enrichir. Si un chercheur souhaite réaliser une nouvelle analyse afin de répondre à une question de recherche particulière, il peut construire un corpus d'analyse et extraire les données à analyser en utilisant les scripts développés. Nous présentons dans les deux sections suivantes le corpus initial « COO-POO », suivi d'exemples d'analyses partagées dans un corpus d'analyse.

9.5 Corpus COO-POO

Dans cette section, nous présentons le corpus COO-POO relatif à l'expérimentation menée à l'IUT de Chambéry dans le cadre du module « Conception orientée-objet, Programmation orientée-objet » en utilisant la plateforme d'apprentissage Moodle (Moodle, 2013). Nous commençons par la présentation de l'environnement Moodle, et de l'expérimentation ayant produit le corpus. Nous décrivons ensuite la composition du corpus. Enfin, nous donnons des exemples de scripts d'interrogation du corpus permettant l'extraction des données relatives aux interactions dans le forum de discussion.

9.5.1 Présentation de l'environnement Moodle

Moodle (Modular Object-Oriented Dynamic Learning Environment) (Moodle, 2013) est un environnement d'apprentissage virtuel. Moodle est open source et est très utilisé partout dans le monde par de nombreuses universités, notamment françaises. Il permet aux enseignants de créer des cours en offrant une multitude de modules. Moodle peut être utilisé par un enseignant pour mettre des ressources pédagogiques à la disposition des apprenants. Mais Moodle présente tout son intérêt dans le cadre d'un enseignement collaboratif en fournissant la possibilité aux participants de partager et communiquer entre eux via des modules tels que le « forum », le « chat », et le « wiki ». Un enseignant peut aussi utiliser Moodle dans l'évaluation de ses étudiants via les modules « test » et « dépôt de fichier ».

L'IUT de Chambéry a installé la plateforme Moodle afin de permettre aux enseignants de créer des cours en ligne afin de mettre des ressources de cours à la disposition des étudiants et d'améliorer le travail collaboratif en utilisant les différents modules de communication offerts par la plateforme. Le corpus décrit dans ce chapitre contient des ressources traces collectées par la plateforme Moodle que l'on a extrait de la base de données associée.

9.5.2 Présentation de l'expérimentation

L'expérimentation « COO-POO » correspond à trois séances de travaux pratiques du module « Conception orientée objet, Programmation orientée objet » en situation d'apprentissage authentique d'un groupe de neuf étudiants en multimédia. L'objectif de cet enseignement était d'approfondir les compétences des étudiants en programmation orientée objet et en langage de programmation Java. Chaque étudiant devait implémenter un téléphone logiciel (softphone) en Java et en utilisant l'environnement de développement intégré Eclipse (Eclipse, 2013). Les étudiants communiquaient entre eux et avec l'enseignant pour poser des questions et demander de l'aide. Le rôle de l'enseignant était clairement défini comme catalyseur et facilitateur via les discussions directes en face à face et les discussions médiatisées via le forum de discussion. Le travail a été réparti sur quatre séances et les étudiants avaient à rendre leurs travaux en cours à la fin de chaque séance. L'observation que nous avons menée a été principalement basée sur l'utilisation de certains outils de la plateforme d'apprentissage Moodle. En outre, les deux dernières séances de travaux pratiques ont été filmées. Au cours de la séance, la communication a été réalisée en face-à-face en utilisant un outil de communication forum offert par la plateforme Moodle, en plus de la

communication humaine directe (dialogue et illustrations sur un tableau blanc classique). Les étudiants ont été encouragés à participer à des discussions et à poser des questions dans le forum. Le reste du temps, c'est à dire entre les séances, le travail a été fait à distance et les étudiants communiquaient avec l'enseignant par l'intermédiaire du forum. À la fin de chaque séance, les étudiants avaient déposé leurs fichiers intermédiaires au moyen du service de dépôt Moodle. L'enseignant avait comme objectif de mesurer le degré de collaboration des étudiants entre eux et entre les étudiants et lui-même. Son but était de faciliter l'apprentissage collaboratif d'investigation. En d'autres termes, les étudiants cherchent l'information auprès des autres étudiants ou de l'enseignant, sachant que le résultat de leurs investigations doit être capitalisé dans le forum. À cet effet, le forum a été structuré en six sujets de discussion. Ces sujets traitent les différents aspects du travail demandé aux étudiants.

9.5.3 Le corpus

Pour construire ce corpus, nous avons collecté les ressources produites durant l'expérimentation décrite ci-dessus. Ce corpus contient une ressource de description RDF contenant les métadonnées de description du corpus et des ressources qui le composent. Ce corpus est composé de quatre types de ressources (cf. Figure 86) parmi les six types de ressources introduits dans le chapitre 4 traitant du modèle de corpus. Le corpus contient une ressource de documentation contenant la description de l'expérimentation qui a donné lieu à ce corpus. Cette ressource permet aux autres chercheurs d'avoir une idée sur le déroulement de l'expérimentation et des objectifs l'ayant motivée. Le corpus contient également sept ressources pédagogiques fournies par l'enseignant et mises à la disposition des étudiants pour leur permettre d'atteindre les objectifs du module. Ces ressources pédagogiques se répartissent comme suit : trois ressources de cours, une ressource contenant le sujet du travail à accomplir, une ressource contenant des consignes de programmation, une ressource tutoriel montrant aux étudiants comment créer une archive dans l'environnement Eclipse, et enfin une ressource contenant la correction du diagramme de classe servant de base au travail de programmation à accomplir. Des ressources de production sont également partagées dans le corpus. Ces dernières sont au nombre de quarante sept et représentent les travaux rendus par les étudiants à l'issue des quatre séances de travaux pratiques via l'outil de dépôt de fichiers de la plateforme Moodle. Une partie de ces ressources est illustrée dans la Figure 86 ci-dessous. Enfin, en ce qui concerne les ressources de type traces, le corpus contient sept ressources non automatiquement traitables relatives aux vidéos enregistrées durant les deux

dernières séances de travaux pratiques, et neuf ressources automatiquement traitables contenant les données relatives aux interactions dans le forum de discussion. L'environnement Moodle de l'IUT de Chambéry stocke les données nécessaires à son fonctionnement dans une base de données MySql. Cette base contient des tables où sont enregistrées des traces d'utilisation des différents modules de la plateforme Moodle. Nous avons écrit des requêtes SQL pour interroger la base de données MySql et avons exporté les résultats dans le format XML. Nous avons donc profité de cette fonctionnalité d'export XML pour importer les ressources telles quelles sans prétraitement. Ces ressources contiennent des informations sur le cours, les utilisateurs, les ressources, et les messages postés dans le forum de discussion.

9.5.1 Interrogation du corpus

Nous avons défini un ensemble de scripts permettant l'interrogation des ressources traces pour extraire les données relatives aux interactions dans le forum. Ces scripts, écrit en XQuery, font partie du modèle opérationnel et permettent l'extraction de données relatives aux concepts définis dans le modèle sémantique. La Figure 87 ci-après illustre des exemples de scripts permettant l'interrogation des concepts relatifs aux interactions de forum, l'extraction des données relatives aux interactions de forum à partir d'un ensemble de ressources et le filtrage de ces données extraites pour ne renvoyer que les interactions relatives à un fil de discussion particulier. Le premier script est un exemple de script d'interrogation du concept simple « estampille temporelle de début » qui représente la date à laquelle un message a été posté. Le deuxième script est un exemple de script d'interrogation du concept complexe « indicateur temporel ». Ce dernier peut être défini comme l'agrégation des concepts « estampille temporelle de début » et « estampille temporelle de dernière modification ». La Figure 88 illustre un extrait du modèle sémantique relatif à la définition du concept complexe « indicateur temporel ». Ce deuxième script fait donc appel au premier script correspondant à l'un des concepts simples qui le constituent. Le troisième script est un exemple de script d'interrogation du concept complexe « interaction dans le forum » composé, entre autres, des concepts « indicateur temporel », « expéditeur du message » et « message de forum ». La Figure 89 illustre un extrait du modèle sémantique relatif à la définition du concept « interaction dans le forum ». Le quatrième script est un exemple de script d'extraction permettant de renvoyer à partir d'un ensemble de ressources les données relatives aux interactions de forum. Le cinquième script est un exemple de script de filtrage

permettant de retourner uniquement les interactions relatives à un fil de discussion particulier dans un forum. Il s'agit de définir une condition de sélection permettant de vérifier la donnée relative au concept « fil de discussion » et ne retourner que les interactions correspondant à un fil de discussion particulier. Tous les scripts données en exemple sont liés à la structure des traces collectées par l'environnement d'apprentissage Moodle. Cette structure est définie sous forme d'un schéma XSD. Nous avons utilisé ces scripts pour extraire les données du corpus « COO-POO » et les analyser en utilisant l'outil d'analyse Tatiana (Dyke, 2009). Nous avons écrit un script de filtrage permettant de mettre en forme les données extraites afin d'être conformes au format d'entrée de Tatiana (display format). Ces analyses sont intégrées au corpus d'analyse présenté dans la section suivante. Nous avons également utilisé deux outils parmi ceux offerts par la plateforme CALICO (CALICO, 2013), après avoir écrit un script de formatage permettant de convertir les données au format XMLForum défini par le projet. La plateforme CALICO fournit des outils Web permettant une meilleure lecture des interactions de forum mais ne permet pas d'enregistrer des ressources en sortie. Nous intégrons donc au corpus, deux captures d'écrans illustrant l'utilisation de CALICO.

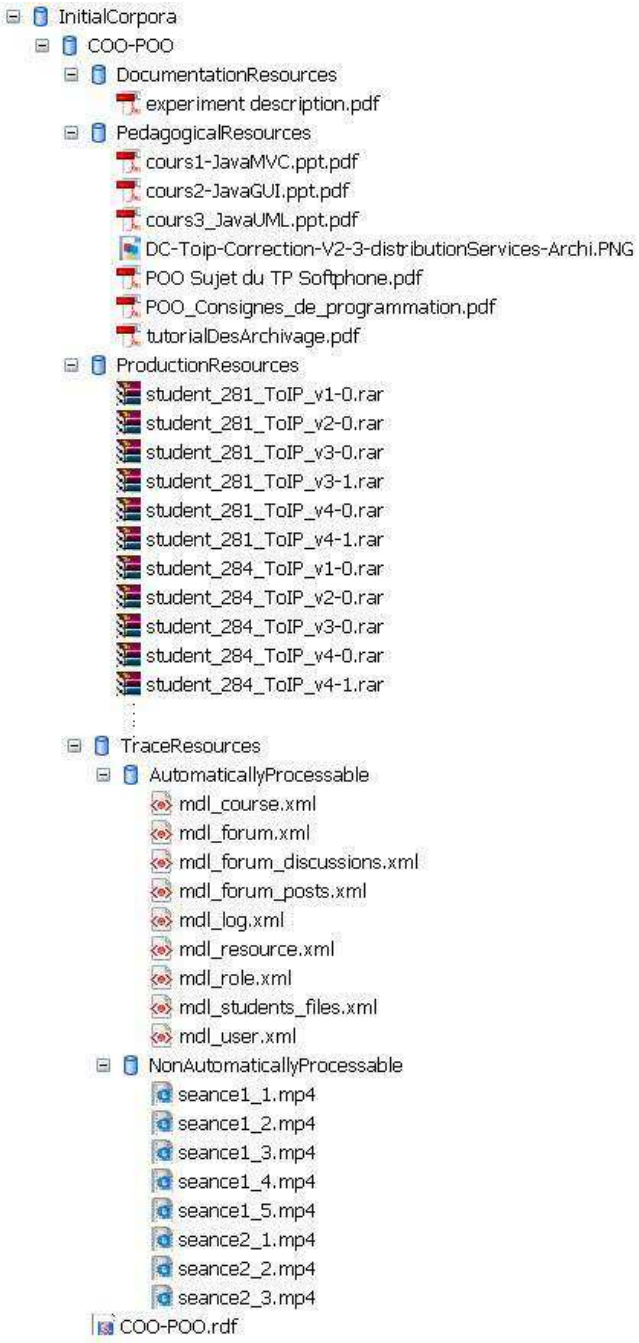


Figure 86 Composants du corpus « COO-POO »

Moodle

Simple concept querying script corresponding to BeginTimestamp concept

```
declare function moodle:forumInteractionBeginTimestamp($postsFile as xs:string)
{
   for $i in doc($postsFile)//table/column[@name="forumPostCreationTime"]/text()
   return $i
};
```

Complex concept querying script corresponding to TemporalIndicator concept

```
declare function moodle:forumInteractionTemporalIndicator($postsFile as xs:string)
{
   let $beginTimestamps := moodle:forumInteractionBeginTimestamp($postsFile)
   let $lastModificationTimestamps := moodle:forumInteractionLastModificationTimestamp($postsFile)
   for $i at $j in $beginTimestamps
   return
   <TemporalIndicator>
   {
      <BeginTimestamp>{$beginTimestamps[$j]}</BeginTimestamp>,
      <LastModificationTimestamp>{$lastModificationTimestamps[$j]}</LastModificationTimestamp>
   }
   </TemporalIndicator>
};
```

Complex concept querying script corresponding to ForumInteraction concept

```
declare function moodle:forumInteraction($postsFile as xs:string, $discussionFile as xs:string,
$usersFile as xs:string,$rolesFile as xs:string)
{
   let $senders := moodle:forumMessageSender($postsFile, $usersFile, $rolesFile)
   let $messages := moodle:forumMessage($postsFile, $discussionFile)
   let $temporalIndicators := moodle:forumInteractionTemporalIndicator($postsFile)
   for $i at $j in $messages
   return
   <ForumInteraction>
   {
      $temporalIndicators[$j],
      $senders[$j],
      $messages[$j]
   }
   </ForumInteraction>
};
```

Exrtracting script retrieving all forum interactions, calls ForumInteraction querying script

```
declare function moodle:forumInteractions($posts as xs:string*, $discussionFile as xs:string*,
$usersFile as xs:string*, $rolesFile as xs:string*)
{
    for $i at $j in $posts
    return moodle:forumInteraction($posts[$j], $discussionFile[$j], $usersFile[$j], $rolesFile[$j])
};
```

Filtering script retrieving forum interactions corresponding to a particular discussion thread, and executed on the output of forumInteractions extracting script

```
declare function moodle:forumInteractionsByThread($posts as xs:string*, $discussionFile as
xs:string*, $usersFile as xs:string*,
$rolesFile as xs:string*, $thread as xs:string)
{
    for $i at $j in $posts
    let $forumInteractions := moodle:forumInteractions($posts[$j], $discussionFile[$j],
          $usersFile[$j], $rolesFile[$j])
    for $f in $forumInteractions
    return
    if($f/ForumMessage/ForumThread/text()[.=$thread])
    then
    $f
    else
    ()
};
```

Figure 87 Exemples de (1) scripts d'interrogation de concept simple et complexe, (2) script d'extraction, et (3) script de filtrage, relatifs aux traces d'interactions de forum de Moodle

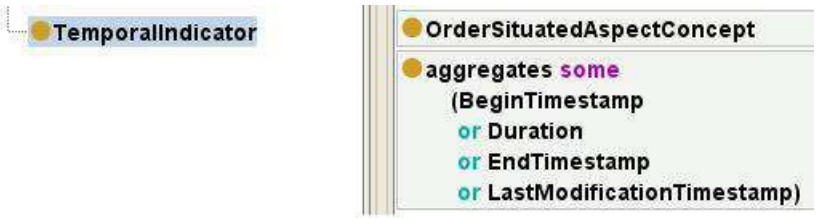


Figure 88 Extrait du modèle sémantique : définition du concept complexe « indicateur temporel » comme l'agrégation des concepts simples « estampille temporelle de début », « durée », « estampille temporelle de fin », et « estampille temporelle de la dernière modification »

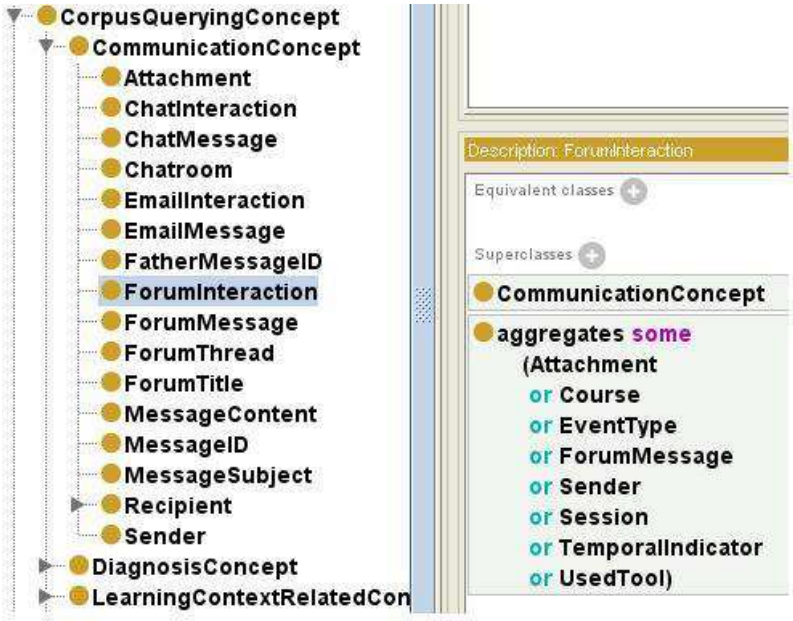


Figure 89 Extrait du modèle sémantique : définition du concept complexe « interaction de forum » comme l'agrégation de concepts complexes et de concepts simples

9.6 Corpus d'analyse

La Figure 90 ci-dessous illustre la composition du corpus. Ce dernier contient une ressource de description RDF contenant les métadonnées de description du corpus et des ressources qui le composent. Par ailleurs, le corpus est composé d'une ressource de documentation offrant une description des analyses réalisées dans le corpus et de la question de recherche étudiée. Par ailleurs, le corpus est composé de onze ressources d'analyse dont huit ressources produites par l'outil d'analyse TATIANA et pouvant être réutilisées par d'autres chercheurs pour les consulter en utilisant l'outil Tatiana, et deux captures d'écran illustrant l'utilisation des outils de la plateforme CALICO. La onzième ressource contient nos

interprétations liées aux analyses réalisées. Ces interprétations peuvent être consultées par d'autres chercheurs intéressés.

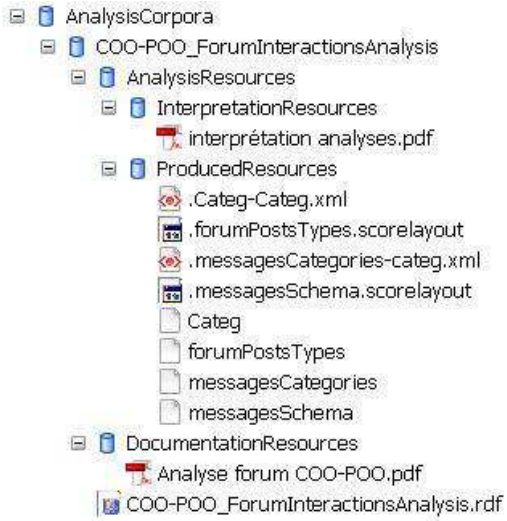


Figure 90 Composants du corpus d'analyse des interactions de forums contenus dans le corpus « COO-POO »

9.6.1 Description des analyses

La construction de ce corpus a pour but d'étudier la question de recherche suivante : « à quel point l'utilisation du forum de discussion a-t-elle amélioré la collaboration étudiant/étudiant et étudiant/enseignant, et contribué à la capitalisation des informations ? ». Pour ceci, nous avons interrogé le corpus « COO-POO » pour extraire les données relatives aux interactions de forum pour les analyser à l'aide de l'outil d'analyse Tatiana et des outils d'analyse / visualisation de la plateforme CALICO. Tatiana étant accessible gratuitement, les chercheurs souhaitant réutiliser ces analyses peuvent télécharger les ressources d'analyse fournies dans ce corpus pour les visualiser dans Tatiana. Celles réalisés dans CALICO sont également facilement reproductibles.

Pour extraire les données relatives au concept « interaction de forum », les différents types de scripts nécessaires à l'extraction doivent être précédemment définis (la Figure 87 illustre une partie de ces scripts). Ces derniers permettent d'extraire les données à partir des corpus. Ensuite, pour utiliser Tatiana, nous avons écrit un script de formatage permettant de formater les données extraites du corpus « COO-POO » afin d'être analysable dans Tatiana. Ce script est illustré dans la Figure 91 ci-dessous. L'import des données dans Tatiana, nous a permis d'avoir une visualisation tabulaire des messages échangés dans le forum (cf. Figure 92). Nous avons créé une catégorisation nous permettant d'attribuer une couleur différente à

chaque participant au fil de discussion « création interface graphique java » du forum (cf. Figure 92). Pour visualiser l'interaction entre les apprenants et l'enseignant, nous avons créé un graphe permettant de visualiser la relation « répond à » illustré dans la Figure 93. La Figure 94 illustre une fonctionnalité très intéressante offerte par l'outil Tatiana et qui consiste à synchroniser différents artefacts, dans notre exemple une visualisation tabulaire et une visualisation graphique des messages envoyés dans le forum.

Dans la deuxième analyse, nous nous sommes intéressés à la catégorisation des messages postés dans les différents fils de discussions du forum. Nous avons identifié treize catégories de messages (cf. Figure 95), chacune étant représentée avec une couleur différente. Les treize catégories identifiées sont :

1. poser une question
2. re-poser une question déjà posée
3. répondre à une question avec une explication
4. répondre à une question en donnant un exemple
5. répondre à une question avec une explication et un exemple
6. contredire une réponse
7. commenter une réponse
8. répondre à sa propre question
9. poser une question, et répondre à sa propre question
10. fournir une information
11. initier un fil de discussion (pour traiter d'un sujet particulier)
12. initier un fil de discussion pour donner une information
13. initier un fil de discussion pour poser une question

Nous avons utilisé cette catégorisation couplée avec une représentation graphique (cf. Figure 96) pour visualiser les différents types des messages postés. Cette visualisation permet d'avoir une représentation visuelle des différentes catégories des messages échangés et de leur proportion. Cette représentation ne donne aucune information sur l'ordre de l'échange de ces messages mais uniquement leur catégorie.

Formatting script for Tatiana analysis tool

```
import module namespace
   jj = "http://kumquat.emse.fr/utilitaires"
   at "jjutils.xq" ;
         import module namespace
   util = "http://www.example.org/AnalysisTools/TatianaAnalysisTool"
   at "utils.xq" ;

   <display>
   {

        let $t := $arguments[1]
        let $d := doc($t)//ForumInteraction

        for $i at $j in $d
            return
            <item>
                <info name="src-anchor">
                   <anchor>{
                       <doc>{ $t }</doc>,
                       <path>{jj:build-Path($i)}</path>
                    }</anchor>
                </info>
                <info name="time">
                   <time>
                       <date>{$i/TemporalIndicator/BeginTimestamp/text()}</date>
                      {if(not(empty($i/TemporalIndicator/Duration/text())))
                         then
                         <duration>{$i/TemporalIndicator/Duration/text()}</duration>
                         else ()}
                   </time>
                </info>
               {
               util:recursiveRetrieving($i)
               }
            </item>
   }
   </display>
```

Figure 91 Script de formatage pour l'outil d'analyse Tatiana

De même que pour Tatiana, nous avons également écrit un script de formatage permettant de convertir les données au format XMLForum défini par le projet CALICO. Les outils offerts par cette plateforme permettent une meilleure lecture des interactions de forum. Ces outils sont plus utiles pour faire des analyses statistiques quantitatives sur les interactions. Nous avons toutefois utilisé deux outils offerts par la plateforme CALICO : (1) l'outil Authagora qui permet de calculer des statistiques sur le nombre des messages envoyés par chaque acteur (cf. Figure 97) ; et (2) l'outil Concordagora qui permet de rechercher les occurrences de mots particuliers et affiche les contextes d'utilisation de ces mots (cf. Figure 98).

Les différents scripts définis dans le cadre de ce travail d'analyse peuvent être réutilisés pour interroger d'autres corpus de forums produits par la plateforme Moodle et

éventuellement les analyser avec Tatiana et CALICO. Si un chercheur souhaite utiliser un autre outil d'analyse, il lui suffira de définir un script de formatage pour permettre d'adapter les données au format attendu par l'outil d'analyse. Tous les autres scripts seront alors automatiquement réutilisables.

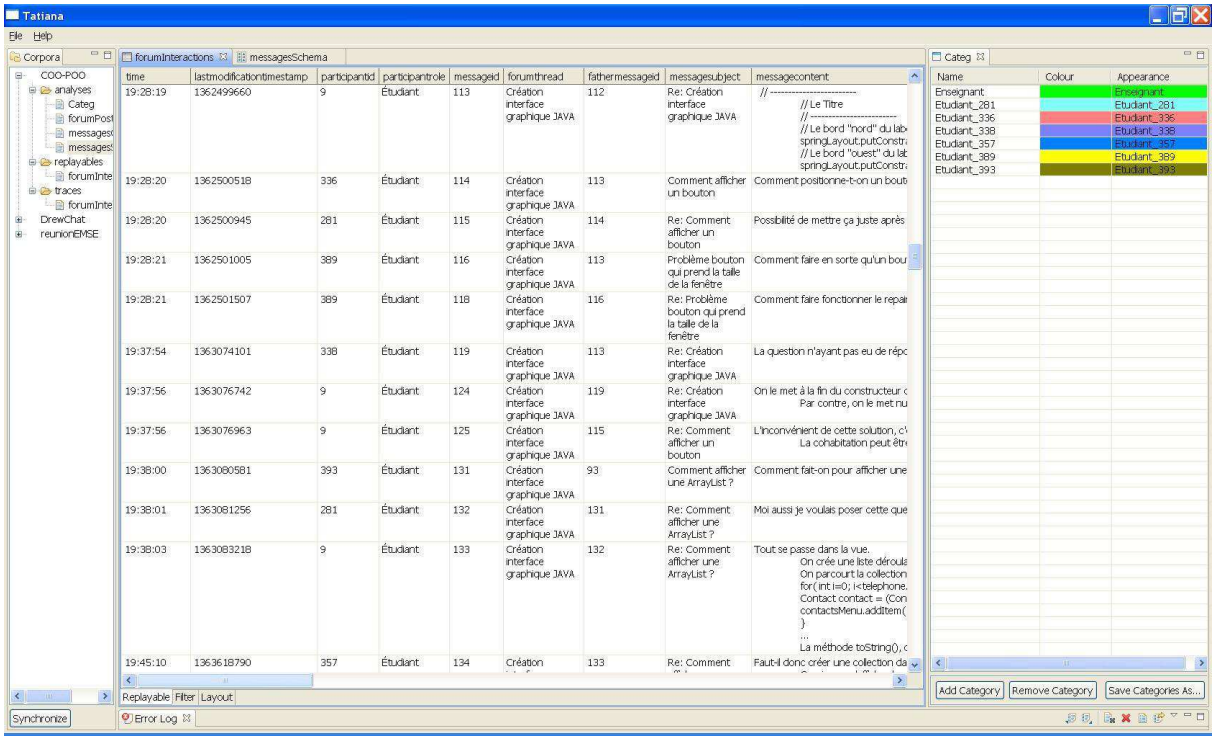


Figure 92 Interface Tatiana, Les interactions de forum du cours « COO-POO » importées dans Tatiana et visualisées sous forme tabulaire (à gauche), et une catégorisation des interactions suivant l'acteur qui en est l'origine

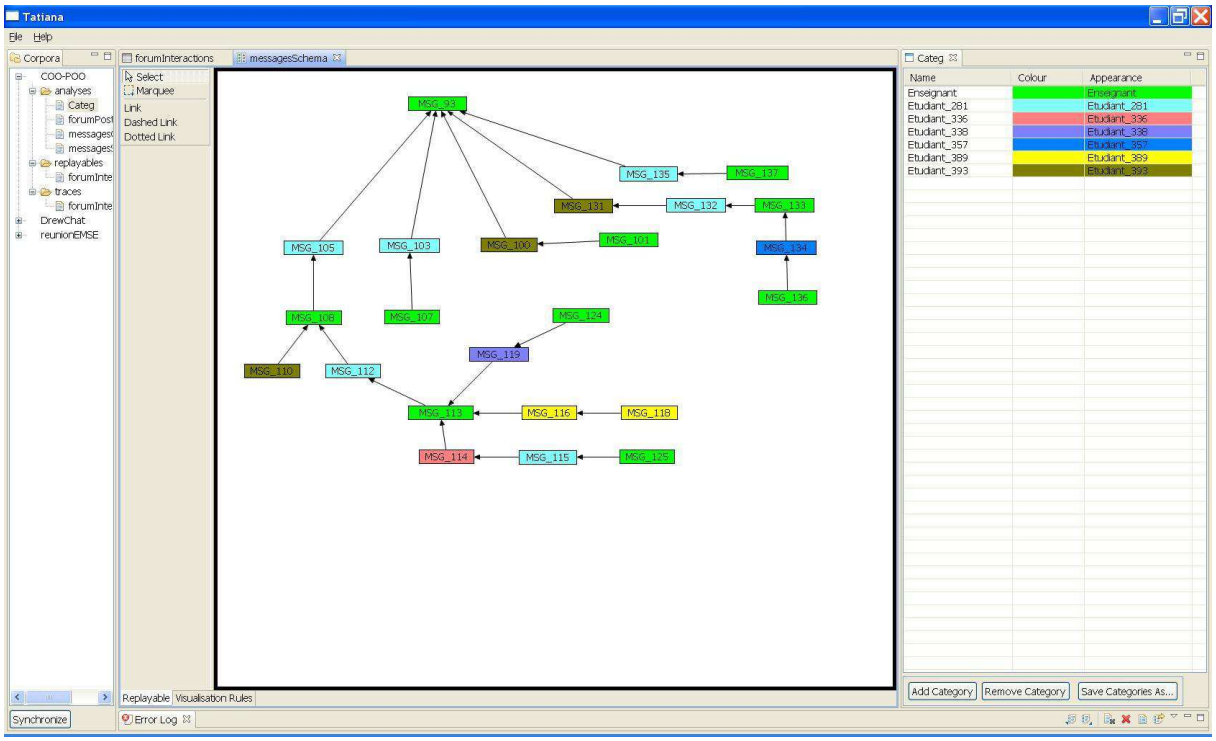


Figure 93 Interface Tatiana, Graphe montrant les différents messages envoyés dans le fil de discussion « création interface graphique java » du forum du cours « COO-POO » ayant une couleur différente pour chaque expéditeur, les messages sont liés entre eux par la relation « répond à »

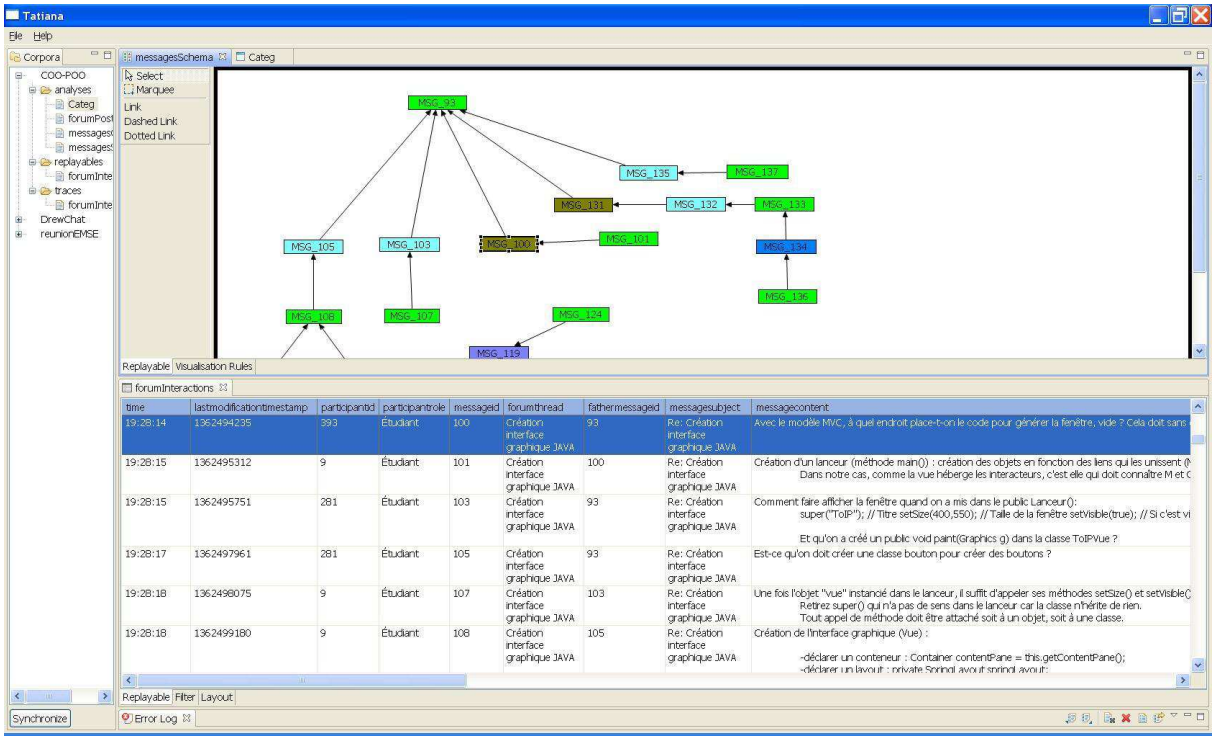


Figure 94 Interface Tatiana, Synchronisation du graphe des relations entre messages avec la visualisation tabulaire

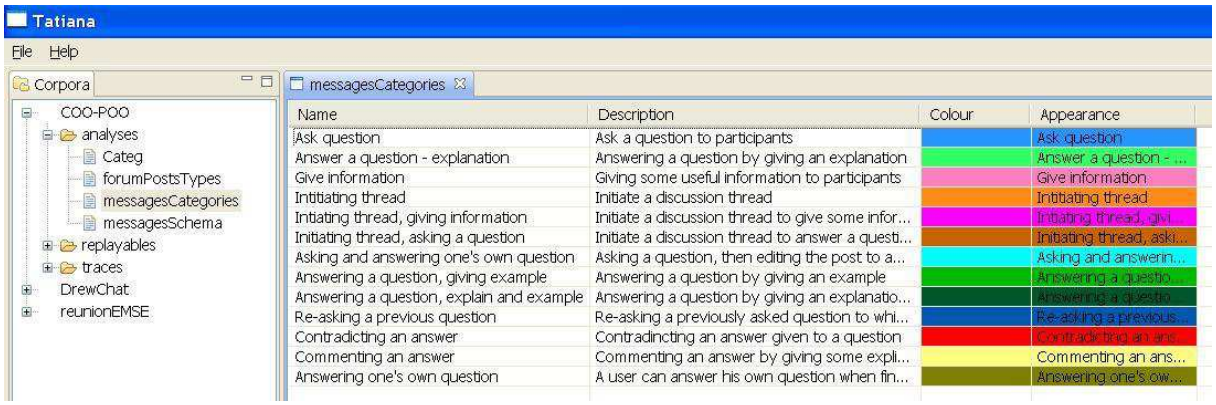


Figure 95 Interface Tatiana, Catégorisation des messages échangées dans le forum du cours « COO-POO »

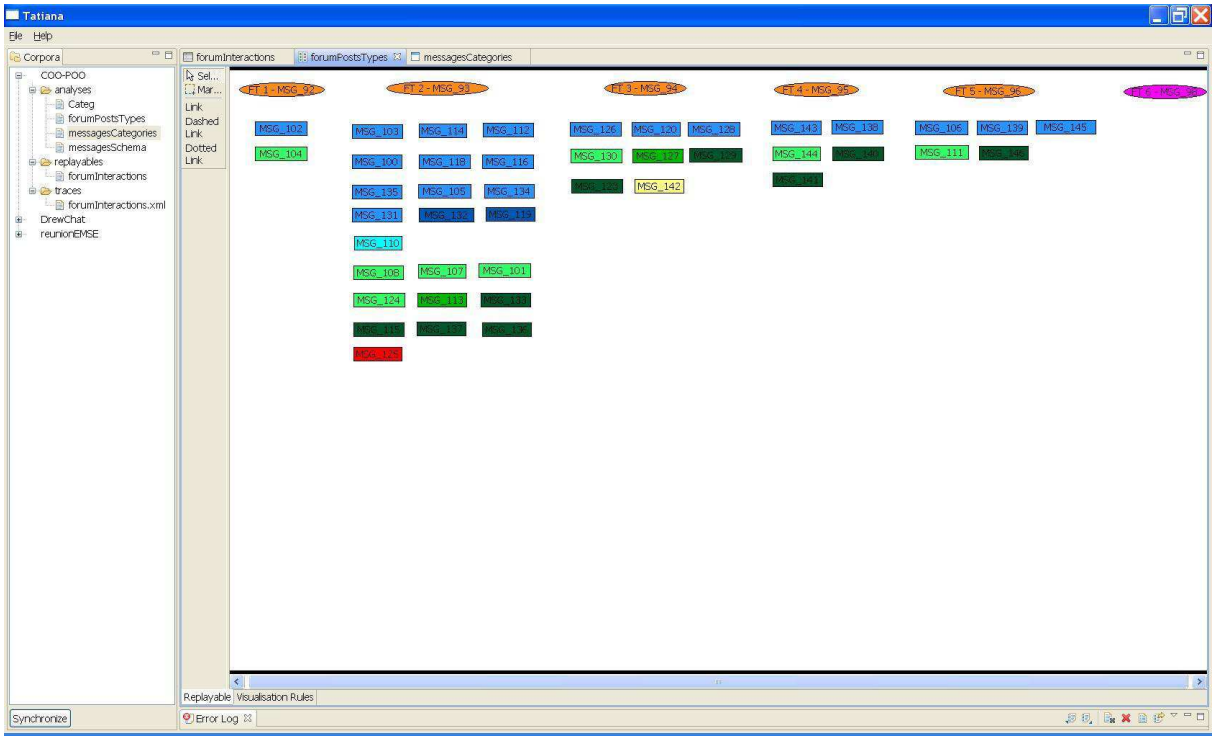
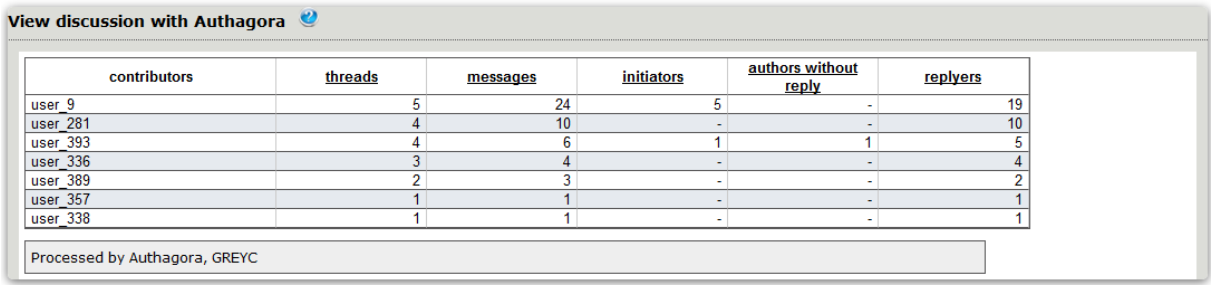


Figure 96 Graphe permettant de visualiser les différents messages échangés dans le forum du cours "COO-POO" coloriés en fonction de leurs catégories


View discussion with Authagora

contributors	threads	messages	initiators	authors without reply	replyers
user_9	5	24	5	-	19
user_281	4	10	-	-	10
user_393	4	6	1	1	5
user_336	3	4	-	-	4
user_389	2	3	-	-	2
user_357	1	1	-	-	1
user_338	1	1	-	-	1

Processed by Authagora, GREYC

Figure 97 Résultat de l'utilisation de l'outil Authagora de la plateforme CALICO sur les traces de forum du corpus « COO-POO »

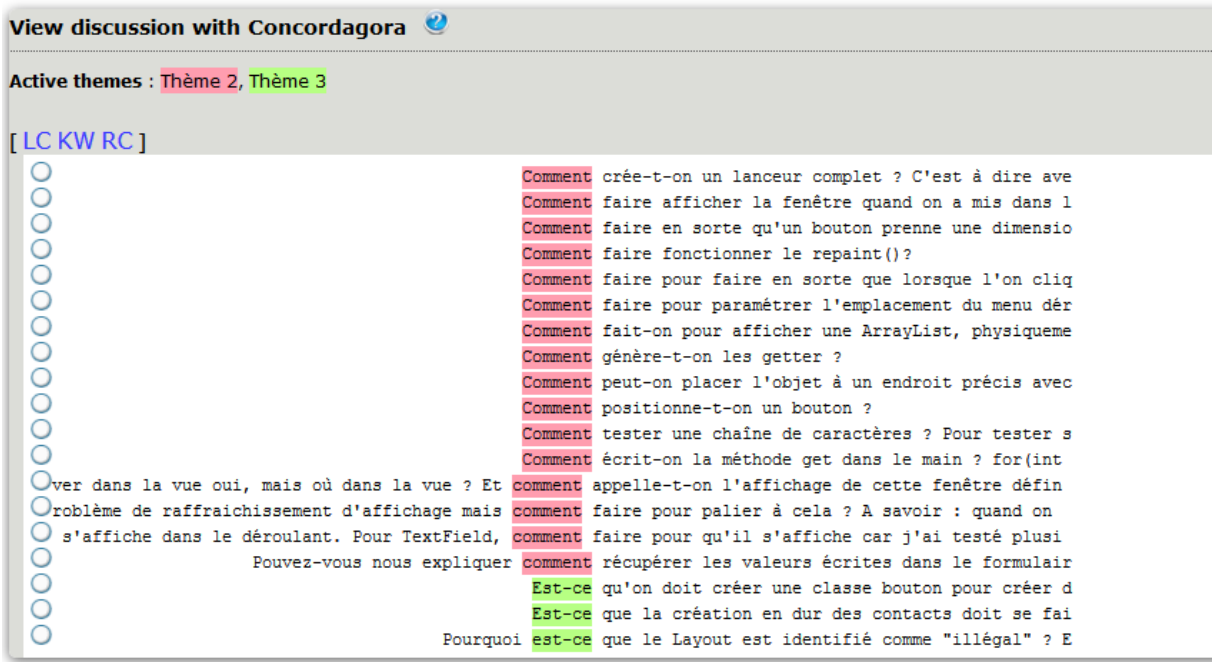


Figure 98 Résultat de l'utilisation de l'outil Concordagora de la plateforme CALICO sur les traces de forum du corpus « COO-POO »

9.6.2 Interprétations

Les analyses réalisées avec Tatiana pour étudier les interactions de forum collectées dans le corpus « COO-POO » nous ont permis de constater que :

1. les étudiants ont utilisé le forum pour poser des questions à l'enseignant, et c'est l'enseignant qui a répondu à toutes les questions posées par les étudiants. Les étudiants n'ont pas répondu aux questions de leurs camarades (à l'exception du seul cas du message dont l'id est 115). Ceci est visible sur la Figure 93 où les messages envoyés par l'enseignant sont colorés en vert. Cela est confirmé par le résultat de l'outil Authagora de CALICO qui montre qu'un utilisateur particulier a envoyé beaucoup plus de messages que les autres utilisateurs

2. L'objectif de l'enseignant concernant la capitalisation des questions permettant aux étudiants de ne pas poser la même question plusieurs fois, a été atteint. Dans deux cas seulement, une même question a été posée deux fois. Dans le premier cas, la question a été posée une deuxième fois car l'enseignant n'a pas répondu à la question. Quant au deuxième cas, l'étudiant a confirmé qu'il se posait la même question qu'un autre étudiant. En utilisant l'outil Concordagora, la recherche des deux termes « Comment » et « est-ce » (cf. Figure 98) nous a

permis d'identifier un ensemble d'interactions relatives à des questions posées dans le forum. L'utilisation de cet outil nous a permis une meilleure lecture plus rapide des interactions.

9.7 Conclusion

Nous avons présenté dans ce chapitre deux exemples de corpus initiaux « EMSE-LEAD » et « COO-POO » construits en se basant sur le modèle de corpus proposé dans le chapitre 4. Nous avons présenté les expérimentations ayant donné lieu à ces corpus, décrit la composition des corpus, et présenté quelques exemples de scripts d'interrogation de ces corpus. Nous avons ensuite présenté un exemple de corpus d'analyse. Ce dernier contient des analyses réalisées sur les données du corpus « COO-POO » relatives aux interactions dans le forum de discussion, et permettant d'étudier la question de recherche « à quel point l'utilisation du forum de discussion a-t-elle amélioré la collaboration étudiant/étudiant et étudiant/enseignant, et contribué à la capitalisation des informations ? ».

Conclusions et perspectives

Ce travail de thèse nous a permis d'apporter des réponses à la problématique de recherche relative au (1) partage de corpus de traces d'interaction d'apprentissage, (2) partage d'outils d'analyse et étude de leur interopérabilité sur les corpus partagés, et (3) liaison entre les corpus partagés et les travaux d'analyse réalisés sur ces corpus. En effet, le partage de corpus collectés suite à des expérimentations écologiques d'apprentissage médiatisé par ordinateur peut être très utile pour un chercheur souhaitant étudier une question de recherche particulière sans être obligé de concevoir et mettre en place une expérimentation. La réutilisation d'un corpus existant lui permet de gagner beaucoup de temps vu la complexité du processus de montage d'une nouvelle expérimentation. La disponibilité de corpus peut aussi servir à un chercheur à éprouver l'applicabilité, l'adaptabilité et la généricité de son outil d'analyse. L'utilité du partage des corpus de traces est renforcée par la possibilité de partager des outils d'analyse permettant l'exploitation des corpus partagés. Un chercheur devient ainsi plus motivé à partager son corpus, utiliser les outils d'analyse partagés, partager son outil d'analyse, son expérience d'analyse, ainsi que les résultats de ses analyses.

Notre étude de l'existant nous a révélé le nombre réduit de travaux s'intéressant à la question de partage de corpus de traces d'interactions et d'outils d'analyse des corpus. Par ailleurs, les travaux existants partagent une même approche pour adresser cette problématique. Cette approche consiste en la proposition d'une structuration des données devant être respectée par les chercheurs souhaitant partager leurs corpus ou utiliser des outils d'analyse partagés (p. ex. tutor message format du projet PSLC Datashop et common format du projet IA JEIRP Kaleidoscope). Cette approche est intuitive puisqu'il suffit de partager une représentation des données pour pouvoir les analyser avec différents outils d'analyse conçus pour accepter cette même représentation. L'inexistence d'une représentation standard des traces d'interaction d'apprentissage a motivé les chercheurs à proposer des représentations. Nous avons à notre tour, commencé à réfléchir à une solution semblable, mais nous avons été confrontés aux limites, à savoir la difficulté à trouver un consensus entre un format de représentation générique pouvant s'adapter à différents besoins de représentation et un format moins générique mais permettant des traitements automatiques des données.

Nous proposons l'approche « Proxyma » comme une alternative flexible et générique permettant le partage de corpus et d'outils d'analyse sans contraindre les chercheurs à exprimer leurs données dans une représentation particulière. En effet, les données collectées par le chercheur sont collectées telles quelles et le chercheur est appelé à donner un minimum d'informations concernant l'expérimentation, et les ressources collectées. Cette approche s'appuie sur trois modèles : (1) « modèle de corpus » définit les types de ressources pouvant être partagées dans un corpus ainsi qu'un modèle de description d'un corpus et des ressources qui le composent ; (2) « modèle sémantique » définit un ensemble de concepts pouvant être partagés entre chercheurs, ces concepts permettent aux chercheurs d'interroger les corpus en vue de les analyser, ce modèle est défini d'une manière participative, évolutive et incrémentale, et peut donc évoluer en fonction des besoins d'analyse des chercheurs ; (3) « modèle opérationnel » définit un ensemble d'opérations permettant un alignement entre les données des corpus collectés et les concepts du modèle sémantique d'un côté, et entre les concepts du modèle sémantique et les formats d'entrée des outils d'analyse de l'autre. Les trois modèles de l'approche « Proxyma » et l'architecture de la plateforme « Beatcorp » qui s'appuie sur cette approche nous ont permis d'apporter des réponses aux trois questions de recherche étudiées. En effet, le modèle de corpus que nous proposons permet de partager des corpus de traces d'interaction contextualisées, et de lier et intégrer les analyses réalisées aux corpus. Par ailleurs, le couple <<modèle sémantique, modèle opérationnel>> permet le partage d'outils d'analyse et leur utilisation pour l'exploitation des corpus partagés en tentant de définir une sémantique partagée par les chercheurs complétée par un mécanisme opérationnel permettant l'alignement de cette sémantique avec les contenus des corpus et les données attendues à l'entrée d'un outil d'analyse.

En réfléchissant à une solution à la problématique étudiée dans ce travail de thèse, nous avions comme objectif de proposer une solution qui soit utilisable par un chercheur le plus rapidement possible, c'est-à-dire sans qu'il ait besoin de faire un grand travail préalable. L'approche utilisée par les autres projets nécessite du chercheur un travail de conversion préalable lui permettant d'importer ses données dans le format commun proposé et de pouvoir utiliser les outils d'analyse partagés. Le fait de garder les données collectées par les chercheurs et de ne pas les convertir nous permet aussi de préserver la richesse sémantique des données initiales qui risque d'être altérée suite à une conversion. Notre approche permet de retarder le travail d'écriture des scripts, réalisant l'alignement entre les données d'un corpus, les concepts du modèle sémantique et les données attendues à l'entrée d'un outil

d'analyse, jusqu'à ce que le chercheur en exprime le besoin. En effet, le développement de ces scripts n'est pas systématique au moment de la construction du corpus puisqu'on n'exige pas de convertir les données dans une représentation particulière. Les chercheurs souhaitant faire une analyse vérifient si les scripts nécessaires à l'extraction des données existent, dans le cas contraire ils écrivent uniquement les scripts dont ils ont besoin. Ces derniers seront accessibles par d'autres chercheurs souhaitant les réutiliser. Nous avons identifié cinq perspectives pour ce travail allant dans le même sens de la facilitation du travail du chercheur souhaitant utiliser la plateforme « Beatcorp » de partage et d'analyse de corpus de traces. En effet, les environnements informatiques pour l'apprentissage humain sont utilisés par des chercheurs de différentes disciplines qui ne possèdent pas toujours des compétences techniques en informatique.

- **Utilisation des données existantes pour deviner et faire des propositions pour minimiser le travail de saisie des utilisateurs :**

En se basant sur les données partagées dans le système, ce dernier pourrait faciliter le travail de saisie d'un chercheur souhaitant construire un corpus. Par exemple, supposons qu'un environnement d'apprentissage ayant déjà servi dans une expérimentation et dont les traces ont déjà été collectées est réutilisé dans une autre expérimentation dont les ressources collectées vont être utilisées pour construire un nouveau corpus. Si le système dispose d'une ressource définissant la structure d'une ressource trace produite par l'environnement d'apprentissage et qui peut être vérifiée automatiquement telle qu'un schéma XSD ou une DTD, le système devrait être capable de détecter que l'environnement d'apprentissage en question est connu et qu'éventuellement un nombre de scripts sont déjà définis pour cette structure et peuvent être réutilisés. Ce type de fonctionnalités peut être très utile pour détecter automatiquement des informations pouvant être utiles pour l'utilisateur sans que ce dernier ait besoin d'interroger le système.

- **Un système de documentation des scripts :**

Afin d'améliorer l'utilisabilité de la plateforme, un système de documentation des scripts permettrait d'améliorer la recherche de scripts par les utilisateurs ou par des modules de la plateforme. En effet, l'interrogation des métadonnées décrivant les scripts existants permet à un chercheur de retrouver facilement si un script correspondant à ses besoins, ou un script similaire, existe. Par ailleurs, un module de fabrication automatique de scripts (voir

point suivant) peut interroger les métadonnées de description des scripts existants pour vérifier l'existence d'un script qu'il souhaite créer. Une telle recherche peut également permettre de retrouver un script utile dans la création automatique d'un nouveau script.

- **Fabrication automatique des scripts :**

Cette deuxième perspective s'intéresse à la possibilité de profiter de la sémantique définie par l'ontologie pour permettre l'automatisation de l'écriture de certains scripts. En effet, nous pouvons envisager une création automatique de scripts d'interrogation et d'extraction se basant sur la sémantique définie par l'ontologie couplée à un dictionnaire de synonymes. Ce qui permettrait d'identifier des données dans les corpus et de construire des scripts automatiquement pour les proposer à des utilisateurs ayant des connaissances informatiques leur permettant de valider ou non les scripts proposés. Un autre exemple intuitif pourrait concerner l'écriture d'un script d'interrogation d'un concept complexe à partir des scripts d'interrogation des concepts qui le constituent. En effet, en se basant sur la définition du concept dans le modèle sémantique défini dans l'ontologie une telle automatisation est envisageable permettant de composer un script à partir d'autres scripts existants.

- **Un système de gestion de cache pour optimiser les temps de réponses :**

L'exécution de requêtes dans la plateforme de partage peut être coûteuse en terme de temps de réponse vu la structure arborescente des documents XML surtout si les données interrogées sont volumineuses. Un système de gestion de cache peut-être mis en place pour utiliser l'historique des requêtes déjà exécutées afin de donner des réponses plus rapides aux requêtes similaires.

- **Langage graphique de script destiné aux utilisateurs non informaticiens :**

La plateforme « Beatcorp » étant destinée à des chercheurs de disciplines différentes qui ne sont pas censés avoir des compétences techniques en informatique leur permettant d'écrire eux même les scripts, et afin d'améliorer l'utilisabilité de la plateforme, nous imaginons qu'un langage graphique assistant les utilisateurs pour créer eux-mêmes leurs scripts peut être d'une grande utilité. La définition d'un tel langage suppose l'existence d'un langage de niveau plus bas définissant une multitude de scripts pré-programmés et appelés par le langage graphique. La Figure 99 ci-dessous illustre une maquette d'un langage graphique et un exemple permettant de construire des scripts en utilisant d'autres existants sans avoir besoin d'écrire du

code. Dans l'exemple, l'utilisateur souhaite interroger deux ressources R1 et R2 d'un corpus, afin de répondre à la requête : « retourner les interactions de chat relatives à l'envoi de message, dont la longueur est supérieure ou égale à 21 caractères et contenant la chaîne de caractères 'program' et convertir les données résultantes pour les analyser dans Tatiana ». Un premier script (« DB-Access ») reçoit en paramètres les ressources à interroger et le script d'extraction à exécuter. Les données résultant de l'exécution du script d'extraction sont passées à différentes instances d'un script de filtrage (« Filter ») qui va opérer sur diverses conditions (p. ex. type événement : envoi message, longueur message >=21, etc.). Une fois les données filtrées, il ne reste qu'à exécuter le script de conversion permettant de préparer les données à l'entrée d'un outil d'analyse (Tatiana dans l'exemple). Il ne reste qu'à enregistrer ces données converties dans un fichier prêt à être analysé (« convert » puis « FileWrite »).

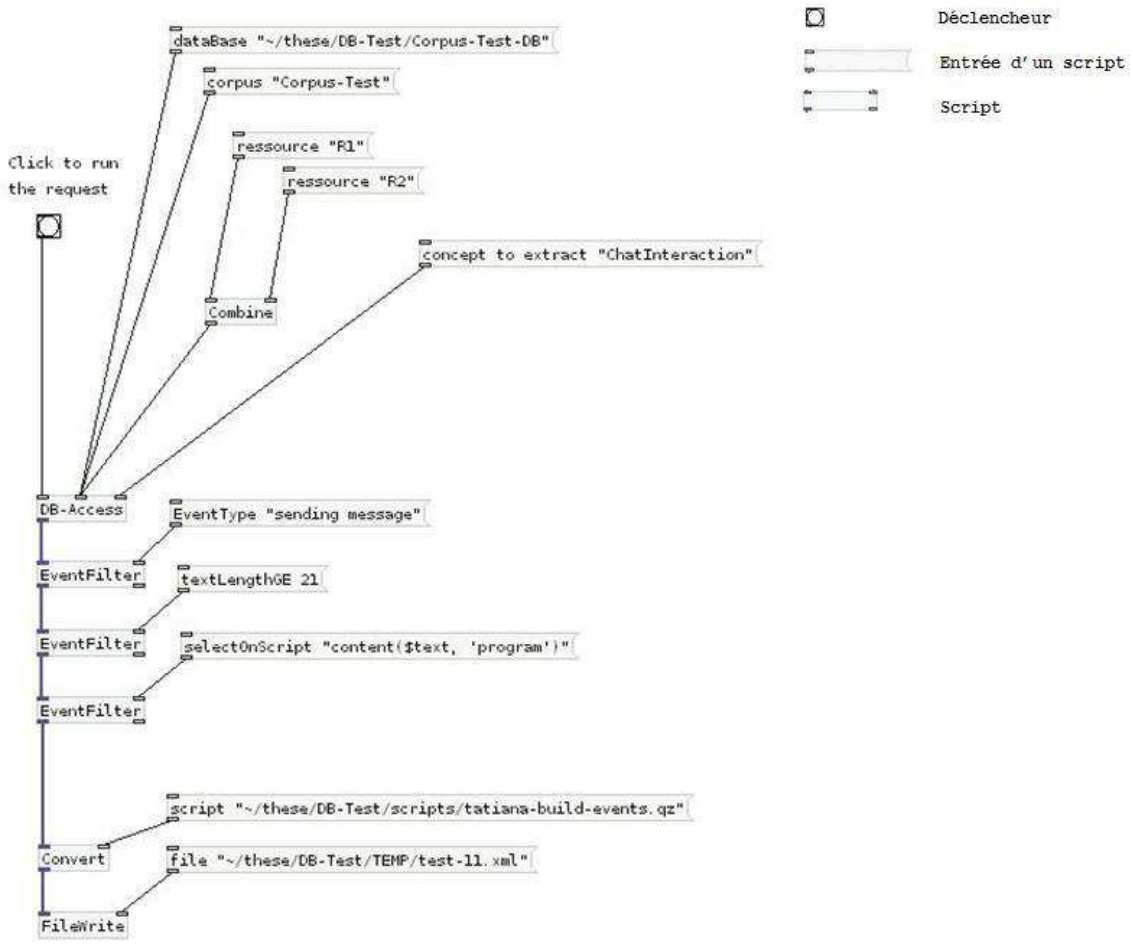


Figure 99 Maquette d'un langage graphique permettant de composer des scripts, exemple répondant à la requête « retourner les interactions de chat relatives à l'envoi de message, dont la longueur est supérieure ou égale à 21 caractères et contenant la chaîne de caractères 'program' et convertir les données pour les analyser dans Tatiana ».

Annexe : État de l'existant

Introduction

Le développement d'un prototype de système mettant en œuvre l'approche « Proxyma » et permettant la manipulation des différents composants de la plateforme « Beatcorp » et l'automatisation des traitements d'un « proxy » n'a pas pu être réalisé dans le cadre de cette thèse. Nous voyons néanmoins comment un tel système peut être développé en se basant sur l'architecture de plateforme et les maquettes présentées dans le chapitre 8 et sur les composants et les scripts dont nous disposons et qui ont été construits durant ce travail de thèse. Nous présentons dans la suite de cette annexe les composants qui existent et qui nous serviront dans le développement du prototype du système. Nous étudions ensuite la réalisation du système en exposant les fonctionnalités qui doivent être développées. Nous présentons ensuite une liste des parties d'un manuel utilisateur de la plateforme. Enfin, nous donnons une vision d'un développement collaboratif autour de la plateforme.

Ce qui existe

Dans le but de valider l'approche « Proxyma », nous avons utilisé eXist-db un système de gestion de base de données XML natif. eXist-db nous a permis de stocker les corpus construits et d'utiliser le moteur XQuery intégré pour exécuter les scripts que nous avons écrits en XQuery pour interroger les corpus. Nous disposons donc d'une installation de serveur eXist-db qui nous permet de gérer les différents composants de la plateforme « Beatcorp ». La Figure 100 ci-dessous illustre la collection « Beatcorp » avec ses différentes sous-collections. Les collections représentent un type d'organisation semblable aux systèmes de fichiers des systèmes d'exploitation. Une collection peut contenir des fichiers de données (XML ou autres p. ex. un document PDF, un enregistrement audio/vidéo, etc.) mais aussi d'autres collections. La collection « Ontology » permet de stocker l'ontologie OWL formalisant les trois modèles de l'approche « Proxyma ». Les collections « InitialCorpora » et « AnalysisCorpora » contiennent respectivement les collections relatives aux corpus initiaux et corpus d'analyse partagés. Chaque corpus est structuré dans une collection contenant une

description RDF du corpus (conforme au modèle de corpus défini dans l'ontologie) et les différentes ressources qui le composent. La collection « LearningTools » contient une ressource RDF contenant la description des environnements d'apprentissage à l'aide d'un ensemble de métadonnées (p. ex. nom, version, page web, éditeur, etc.), et, pour chaque environnement d'apprentissage ayant donné lieu aux données partagées dans un corpus, une collection contenant des données relatives à : (1) des ressources éventuelles de documentation, (2) la structure des traces générées par l'environnement d'apprentissage (p.ex. DTD ou schéma XSD) et (3) les scripts d'extraction permettant de définir la sémantique des données produites par l'environnement d'apprentissage par rapport aux concepts du modèle sémantique. La collection « AnalysisTools » contient une ressource RDF contenant la description des environnements d'analyse, et, pour chaque outil d'analyse intégré à la plateforme, (1) des ressources éventuelles de documentation, (2) la structure des données attendues à l'entrée de l'outil d'analyse, et (3) les scripts de formatage permettant d'aligner la sémantique de l'outil d'analyse avec celle des données extraites des corpus. La collection « Agents » contient les données concernant les personnes (créateurs et contributeurs) intervenant dans la construction des corpus partagés. La collection « Lib » contient des librairies utiles dans le développement des scripts ainsi qu'un ensemble de scripts génériques utiles pour la conversion des types de données (p. ex. conversions de dates). Les collections « QueryResults » et « Tests » contiennent des ressources relatives aux tests que l'on a effectués pour exécuter des scripts sur les corpus.

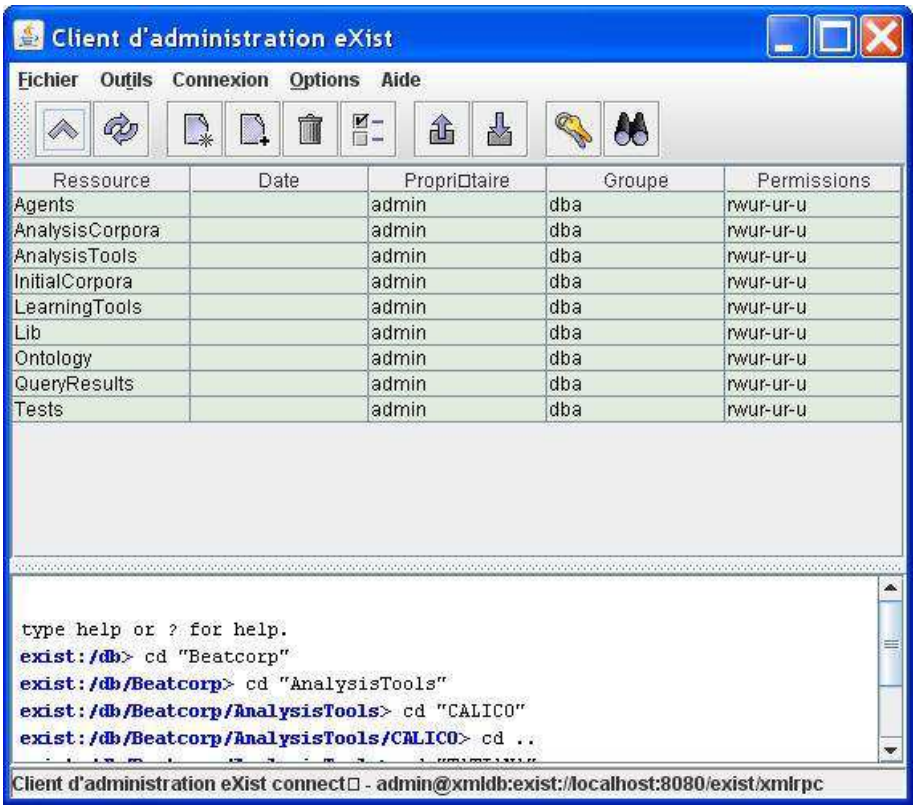


Figure 100 Ecran eXist-db montrant les différentes collections que nous gérons

Base de corpus

Le premier composant de la plateforme géré dans eXist-db est la base de corpus. Cette dernière se compose des trois corpus construits dans le cadre de la validation de l'approche et présentés dans le chapitre 9 : deux corpus initiaux « EMSE-LEAD » et « COO-POO », et un corpus d'analyse « COO-POO_ForumInteractionsAnalysis » contenant des analyses des interactions de forum du corpus « COO-POO » dans le but de répondre à la question de recherche : « à quel point l'utilisation du forum de discussion a-t-elle amélioré la collaboration étudiant/étudiant et étudiant/enseignant, et contribué à la capitalisation des informations ? ».

Ontologie

L'ontologie est le composant central de l'approche « Proxyma » permettant de définir les trois modèles de l'approche : le modèle de corpus, le modèle sémantique et le modèle opérationnel. La structure et la description des corpus partagés dans la base de corpus suivent la définition du modèle de corpus dans l'ontologie. Les concepts interrogeables relatifs à la sémantique des données contenues dans les corpus partagés sont définis dans la partie de l'ontologie relative au modèle sémantique. Et enfin la description des scripts d'interrogation,

conversion, et formatage utilisés dans l'interrogation des corpus suit la définition des scripts du modèle opérationnel défini dans l'ontologie. Ces descriptions seront utilisées par le « proxy » afin d'automatiser les traitements visant à préparer les données à l'entrée d'un outil d'analyse particulier. Pour construire cette ontologie, nous avons utilisé Protégé[16], un éditeur graphique d'ontologie libre d'accès permettant d'exprimer l'ontologie dans le langage OWL (entre autres). L'ontologie construite dans protégé est stockée sur notre serveur eXist-db. Actuellement, nous manipulons l'ontologie dans Protégé, mais dans le cadre d'une application Web permettant de gérer les différents composants de la plateforme, nous pensons qu'une interface permettant la visualisation et la modification de l'ontologie d'une manière intégrée peut présenter un intérêt pour les utilisateurs. La Figure 101 ci-dessous illustre un aperçu de l'ontologie visualisée dans Protégé.

[16] http://protege.stanford.edu

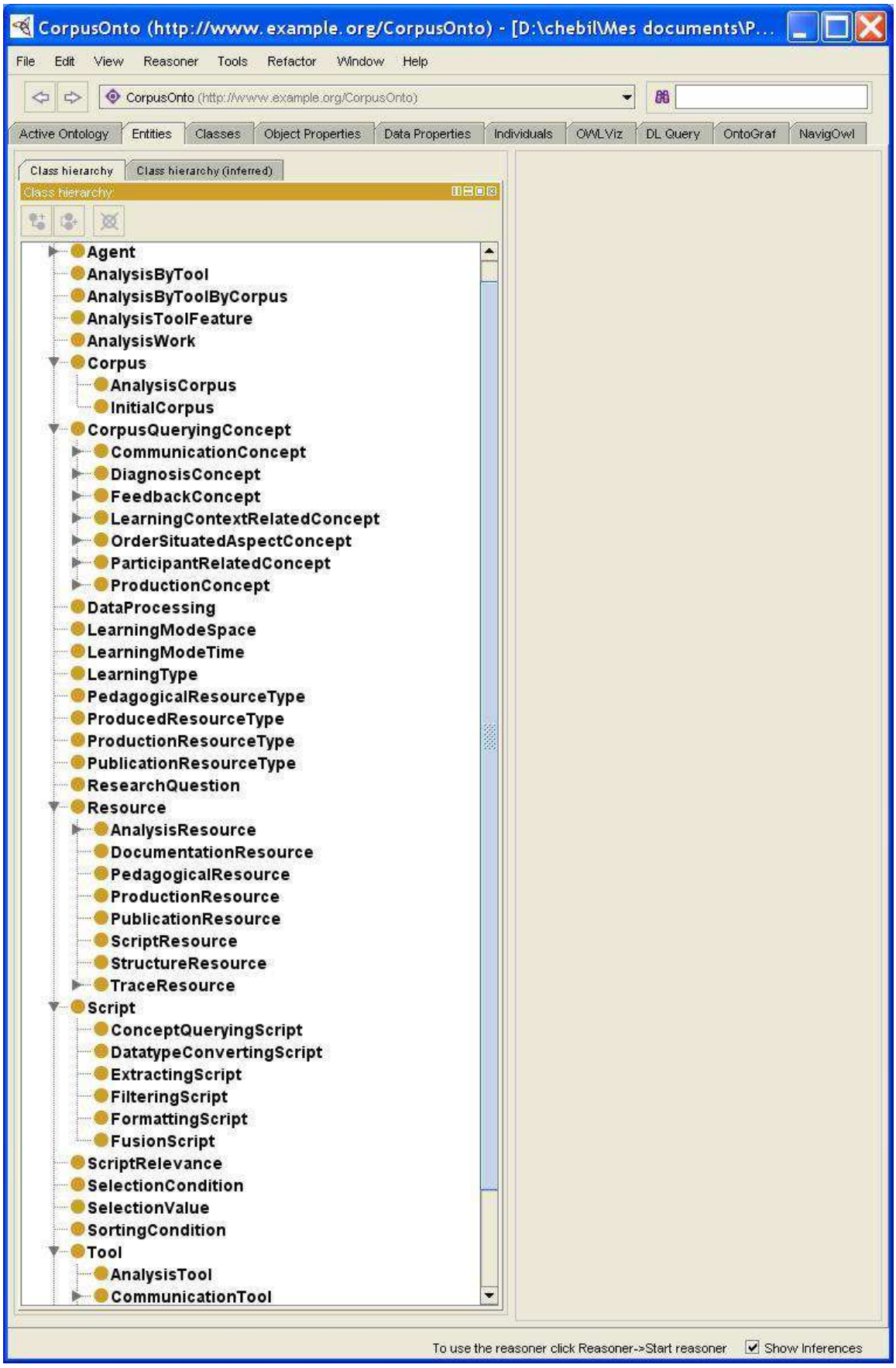


Figure 101 Aperçu de l'ontologie dans l'éditeur Protégé

Base de scripts

Comme nous l'avons déjà introduit, les scripts composant la base de scripts sont répartis sur trois collections : « LearningTools », « AnalysisTools », et « Lib ». Les scripts existants dans la base de scripts correspondent aux exemples d'application présentés dans le chapitre 9. Ces scripts correspondent aux scripts d'interrogation de concepts, scripts de conversion de types de données, scripts d'extraction, scripts de filtrage, et script de formatage développés pour permettre l'extraction des données relatives aux interactions dans le chat de l'environnement DREW (corpus « EMSE-LEAD ») et des interactions dans le forum de la plateforme Moodle (corpus « COO-POO ») et leur conversion pour être analysées dans Tatiana. Nous disposons donc actuellement des scripts suivants :

- les scripts relatifs à l'extraction des interactions relatives aux discussions de chat générées par l'EIAH DREW (cf. Figure 102 et Figure 103) ;
- les scripts relatifs à l'extraction des interactions relatives aux discussions dans le forum de l'EIAH Moodle (cf. Figure 104 et Figure 105);
- deux scripts permettant le formatage des données relatives aux interactions de chat et aux interactions de forum pour l'entrée de Tatiana (cf. Figure 106 et Figure 107) (ce formatage étant lié aux concepts définis dans le modèle sémantique, il n'est pas lié au format des données des corpus, et est donc réutilisable pour des données équivalentes provenant d'environnements d'apprentissage différents) ;
- Un script permettant le formatage des données relatives aux interactions de forum pour permettre leur analyse en utilisant les outils de la plateforme CALICO.

Certains de ces scripts ont déjà été présentés dans le corps du document pour illustrer un point précis de notre description.

```
module namespace drew = "http://www.example.org/Beatcorp/LearningTools/DrewLearningTool";

import module namespace functx = "http://www.functx.com" at
"http://localhost:8080/exist/rest//db/Beatcorp/Lib/functx.xqm";

import module namespace utils = "http://www.example.org/LearningTools/utils"
    at "http://localhost:8080/exist/rest//db/Beatcorp/Lib/utils.xqm" ;

declare function drew:chatMessage($doc as xs:string, $position as xs:integer) as xs:string
{
    let $msg := (doc($doc)//chat)[$position]/text/text()
    let $empty := ""
    return
    if (empty($msg))
    then $empty
    else
    $msg
};

declare function drew:chatMessage($doc as xs:string) as element()*
{
    let $messagesContents := drew:chatMessageContent($doc)
    for $i at $j in $messagesContents
    return
    <ChatMessage>
    {<MessageContent>
    {
    $messagesContents[$j]
    }
    </MessageContent>
    }
    </ChatMessage>
};

declare function drew:chatMessageContent($doc as xs:string) as xs:string*
{
    let $msg := doc($doc)//chat/text
    for $i in $msg
    return
    if (empty($i/text()))
    then ""
    else
    $i/text()
};

declare function drew:chatMessageSenderUsername($doc as xs:string) as xs:string*
{
    doc($doc)//chat/../@user
};


declare function drew:chatMessageSender($doc as xs:string) as node()*
{
    let $sendersUsernames := drew:chatMessageSenderUsername($doc)
    for $i in $sendersUsernames
    return
    <Sender>
    {
    <ParticipantUsername>{$i}</ParticipantUsername>
    }
    </Sender>
};

declare function drew:chatTemporalIndicator($doc as xs:string) as node()*
{
    let $temporalIndicatorBegintimestamps := drew:chatTemporalIndicatorBeginTimestamp($doc)
    let $temporalIndicatorDuration := drew:chatTemporalIndicatorDuration($doc)
    for $i at $c in $temporalIndicatorBegintimestamps
    return
    <TemporalIndicator>
    {
    <BeginTimestamp>{$temporalIndicatorBegintimestamps[$c]}</BeginTimestamp>,
    <Duration>{$temporalIndicatorDuration[$c]}</Duration>
    }
    </TemporalIndicator>
};

declare function drew:chatTemporalIndicatorBeginTimestamp($doc as xs:string) as xs:string*
{
    (doc($doc)//chat)/../time/date/text()
};

declare function drew:chatTemporalIndicatorDuration($doc as xs:string) as xs:string*
{
    (doc($doc)//chat)/../time/duration/text()
};
```

Figure 102 Scripts relatifs aux interactions de chat dans l'environnement d'apprentissage DREW (1/2)

```
declare function drew:chatInteraction($doc as xs:string) as node()*
{
    let $chatMessagesSenders := drew:chatMessageSender($doc), $chatTemporalIndicator := drew:chatTemporalIndicator($doc),
    $chatMessages := drew:chatMessage($doc)
    for $i at $c in $chatMessages
    return
    <ChatInteraction>
    {
    $chatMessagesSenders[$c],
    $chatTemporalIndicator[$c],
    $chatMessages[$c]
    }
    </ChatInteraction>
};

declare function drew:chatInteractionByUser($chatInteractions as node()*, $user as xs:string) as node()*
{
    for $i at $c in $chatInteractions
    return
    if($i/Sender/ParticipantUsername/text()[.=$user])
    then
    $i
    else
    ()
};

declare function drew:chatInteractionOtherUser($chatInteractions as node()*, $user as xs:string) as node()*
{
    for $i at $c in $chatInteractions
    return
    if($i/Sender/ParticipantUsername/text()[.!=$user])
    then
    $i
    else
    ()
};

declare function drew:chatInteractions($docs as xs:string*) as node()*
{
    for $i in $docs
    let $chatInteractions:=drew:chatInteraction($i)
    for $j at $c in $chatInteractions
    return
    <ChatInteraction>
    {
        $chatMessagesSenders[$c],
        $chatMessages[$c]
    }
    </ChatInteraction>
};

declare function drew:chatInteractionMessageLengthGE($chatInteractions, $length) as node()*
{
    for $i in $chatInteractions
    return
    if(fn:string-length($i/ChatMessage/MessageContent/text())>=$length)
    then
    $i
    else
    ()
};

declare function drew:chatInteractionMessageContainsString($chatInteractions, $expression) as node()*
{
    for $i in $chatInteractions
    return
    if(fn:contains($i/ChatMessage/MessageContent/text(), $expression))
    then
    $i
    else
    ()
};

declare function drew:chatMessageSendingInteractions( $chatInteractions as node()* ) as node()*
{
        for $i at $j in $chatInteractions
        return
        if(empty($i/ChatMessage/MessageContent/text()))
        then ()
        else $chatInteractions[$j]
};

declare function drew:chatUser($doc as xs:string, $position as xs:integer) as xs:string
{
    (doc($doc)//chat)[$position]/../@user
};

declare function drew:chatBeginDate($doc as xs:string, $position as xs:integer) as xs:integer
{
    (doc($doc)//chat)[$position]/../time/date/text()
};


declare function drew:chatDuration($doc as xs:string, $position as xs:integer) as xs:integer?
{
    (doc($doc)//chat)[$position]/../time/duration/text()
};

declare function drew:chatEndDate($doc as xs:string, $position as xs:integer) as xs:integer
{
    if(not(empty(drew:chatDuration($doc, $position))))
    then
    drew:chatBeginDate($doc,$position) + drew:chatDuration($doc, $position)
    else
    drew:chatBeginDate($doc,$position)
};

declare function drew:allChatMessageSendingInteractions( $allChatInteractions as node()* ) as node()*
{
        for $i at $j in $allChatInteractions
        return
        if(empty($i/Message/text()))
        then ()
        else $allChatInteractions[$j]
};
```

Figure 103 Scripts relatifs aux interactions de chat dans l'environnement d'apprentissage DREW (2/2)

```
module namespace moodle = "http://www.example.org/LearningTools/MoodleLearningTool";

import module namespace functx = "http://www.functx.com" at
"http://localhost:8080/exist/rest//db/Beatcorp/Lib/functx.xqm";

declare function moodle:forumMessageSenderID($postsFile as xs:string)
{
   for $i in doc($postsFile)//table/column[@name="forumPostSender"]/text()
   return $i
};

declare function moodle:forumMessageSenderRole($sendersIDs as xs:string*, $usersFile as xs:string,
$rolesFile as xs:string)
{
   for $i in $sendersIDs
   let $roleID:=doc($usersFile)//table/column[@name="userId"]/text()[. =
$i]/../../column[@name="userRoleId"]/text()
   let $roleName :=
doc($rolesFile)//table/column[@name="roleId"]/text()[.=$roleID]/../../column[@name="roleName"]/text()
return
$roleName
};

declare function moodle:forumMessageSender($postsFile as xs:string, $usersFile as xs:string,
$rolesFile as xs:string)
{
  let $messagesSendersIds := moodle:forumMessageSenderID($postsFile)
  let $messagesSendersRoles :=
moodle:forumMessageSenderRole($messagesSendersIds,$usersFile,$rolesFile)
  for $i at $j in $messagesSendersIds
  return
  <Sender>
  {
<ParticipantID>{$messagesSendersIds[$j]}</ParticipantID>,
<ParticipantRole>{$messagesSendersRoles[$j]}</ParticipantRole>
  }
  </Sender>
};

declare function moodle:forumMessageID($postsFile as xs:string)
{
   for $i in doc($postsFile)//table/column[@name="forumPostId"]/text()
   return $i
};

declare function moodle:forumMessageThread($postsFile as xs:string, $discussionFile as xs:string)
{
let $discussionId := doc($postsFile)//table/column[@name="forumPostDiscussion"]/text()
   for $i in $discussionId
   let $current:=doc($discussionFile)//table/column[@name="forumDiscussionId"]/text()[.=
$i]/../../column[@name="forumDiscussionName"]/text()
   return
$current
};

declare function moodle:forumMessageSubject($postsFile as xs:string)
{
doc($postsFile)//table/column[@name="forumPostSubject"]/text()

};

declare function moodle:forumMessageContent($postsFile as xs:string)
{

let $messagesContents := doc($postsFile)//table/column[@name="forumPostMessage"]/text()
for $i in $messagesContents
(:let $res :=fn:replace($i, '&lt;','<'),
$res:=fn:replace($res, '&gt;','>'),:)
let $res:=fn:replace($i, '&lt;span class=".\d" style="line-height: \d.\d;"&gt;',''),
$res:=fn:replace($res, '&lt;span style="line-height: \d.\d;"&gt;',''),
$res:=fn:replace($res, '&lt;span style="line-height: \d.\d; color: #\d*.*\d*;"&gt;',''),
$res:=fn:replace($res, '&lt;span style="color: #\d*.*\d*;"&gt;',''),
$res:=fn:replace($res, '&lt;span class=".\d+"&gt;',''),
$res:=fn:replace($res, '&lt;p class=".\d*"&gt;',''),
$res:=fn:replace($res, '&lt;img.*/&gt;',''),
$res:=fn:replace($res, '&lt;p&gt;',''),
$res:=fn:replace($res, '&lt;/p&gt;',''),
$res:=fn:replace($res, '&lt;span&gt;',''),
$res:=fn:replace($res, '&lt;/span&gt;',''),
$res:=fn:replace($res, '&lt;span class="edited"&gt;',''),
$res:=fn:replace($res, '&lt;span style="white-space: pre;"&gt;',''),
$res:=fn:replace($res, '&lt;strong&gt;',''),
$res:=fn:replace($res, '&lt;/strong&gt;',''),
$res:=fn:replace($res, '&lt;em&gt;',''),
$res:=fn:replace($res, '&lt;/em&gt;',''),
$res:=fn:replace($res, ' ',' '),
$res:=fn:replace($res, '&lt;br /&gt;',''),
$res:=fn:replace($res, '&lt;ol&gt;',''),
$res:=fn:replace($res, '&lt;/ol&gt;',''),
$res:=fn:replace($res, '&lt;li&gt;','-'),
$res:=fn:replace($res, '&lt;/li&gt;',''),
$res:=fn:replace($res, '&amp;lt;','<'),
$res:=fn:replace($res, '&amp;gt;','>')

return $res

};
```

Figure 104 Scripts relatifs aux interactions de forum dans l'environnement d'apprentissage Moodle (1/2)

```
declare function moodle:forumMessageFatherID($postsFile as xs:string)
{
doc($postsFile)//table/column[@name="forumPostParent"]/text()
};

declare function moodle:forumMessage($postsFile as xs:string, $discussionFile as xs:string)
{
let $messagesIds := moodle:forumMessageID($postsFile)
let $messagesThreads := moodle:forumMessageThread($postsFile, $discussionFile)
let $messagesSubjects := moodle:forumMessageSubject($postsFile)
let $messagesContents := moodle:forumMessageContent($postsFile)
let $fathersIds := moodle:forumMessageFatherID($postsFile)
for $i at $j in $messagesIds
return
<ForumMessage>
{
<MessageID>{$messagesIds[$j]}</MessageID>,
<ForumThread>{$messagesThreads[$j]}</ForumThread>,
<FatherMessageID>{$fathersIds[$j]}</FatherMessageID>,
<MessageSubject>{$messagesSubjects[$j]}</MessageSubject>,
<MessageContent>{$messagesContents[$j]}</MessageContent>
}
</ForumMessage>
};

declare function moodle:forumInteractionBeginTimestamp($postsFile as xs:string)
{
   for $i in doc($postsFile)//table/column[@name="forumPostCreationTime"]/text()
   return $i
};

declare function moodle:forumInteractionLastModificationTimestamp($postsFile as xs:string)
{
   for $i in doc($postsFile)//table/column[@name="forumPostModificationTime"]/text()
   return $i
};

declare function moodle:forumInteractionTemporalIndicator($postsFile as xs:string)
{
let $beginTimestamps := moodle:forumInteractionBeginTimestamp($postsFile)
let $lastModificationTimestamps := moodle:forumInteractionLastModificationTimestamp($postsFile)
for $i at $j in $beginTimestamps
return
<TemporalIndicator>
{
<BeginTimestamp>{$beginTimestamps[$j]}</BeginTimestamp>,
<LastModificationTimestamp>{$lastModificationTimestamps[$j]}</LastModificationTimestamp>
}
</TemporalIndicator>
};

declare function moodle:forumInteraction($postsFile as xs:string, $discussionFile as xs:string,
$usersFile as xs:string,
$rolesFile as xs:string)
{
let $senders := moodle:forumMessageSender($postsFile, $usersFile, $rolesFile)
let $messages := moodle:forumMessage($postsFile, $discussionFile)
let $temporalIndicators := moodle:forumInteractionTemporalIndicator($postsFile)
for $i at $j in $messages
return
<ForumInteraction>
{
$temporalIndicators[$j],
$senders[$j],
$messages[$j]
}
</ForumInteraction>
};

declare function moodle:forumInteractions($posts as xs:string*, $discussionFile as xs:string*,
$usersFile as xs:string*,
$rolesFile as xs:string*)
{
    for $i at $j in $posts
    return moodle:forumInteraction($posts[$j], $discussionFile[$j], $usersFile[$j], $rolesFile[$j])
};

declare function moodle:forumInteractionsByThread($posts as xs:string*, $discussionFile as
xs:string*, $usersFile as xs:string*,
$rolesFile as xs:string*, $thread as xs:string)
{
    for $i at $j in $posts
    let $forumInteractions := moodle:forumInteractions($posts[$j], $discussionFile[$j],
$usersFile[$j], $rolesFile[$j])
    for $f in $forumInteractions
    return
    if($f/ForumMessage/ForumThread/text()[.=$thread])
    then
    $f
    else
    ()
};
```

Figure 105 Scripts relatifs aux interactions de forum dans l'environnement d'apprentissage Moodle (2/2)

```
import module namespace jj = "http://kumquat.emse.fr/utilitaires" at "jjutils.xq" ;
import module namespace
util = "http://www.example.org/AnalysisTools/TatianaAnalysisTool"
at « utils.xq" ;


<display>
{

let $t := $arguments[1]
let $d := doc($t)//ChatInteraction


for $i at $j in $d
    return
    <item>
        <info name="src-anchor">
                        <anchor>{
                        <doc>{ $t }</doc>,
                        <path>{jj:build-Path($i)}</path>
                        }</anchor>
                </info>
        <info name="time">
                <time>
                    <date>{$i/TemporalIndicator/BeginTimestamp/text()}</date>
                    {if(not(empty($i/TemporalIndicator/Duration/text())))
                    then
                    <duration>{$i/TemporalIndicator/Duration/text()}</duration>
                    else ()}

                </time>
            </info>

       {
       util:recursiveRetrieving($i)
       }
    </item>
}
</display>
```

Figure 106 Script de formatage des données relatives au concept « interaction de chat » du modèle sémantique pour l'entrée de l'outil Tatiana

```
import module namespace jj = "http://kumquat.emse.fr/utilitaires" at "jjutils.xq" ;
import module namespace
util = "http://www.example.org/AnalysisTools/TatianaAnalysisTool"
at "utils.xq" ;


<display>
{

let $t := $arguments[1]
let $d := doc($t)//ForumInteraction


for $i at $j in $d
    return
    <item>
        <info name="src-anchor">
                        <anchor>{
                        <doc>{ $t }</doc>,
                        <path>{jj:build-Path($i)}</path>
                        }</anchor>
                </info>
        <info name="time">
                <time>
                    <date>{$i/TemporalIndicator/BeginTimestamp/text()}</date>
                    {if(not(empty($i/TemporalIndicator/Duration/text())))
                    then
                    <duration>{$i/TemporalIndicator/Duration/text()}</duration>
                    else ()}

                </time>
            </info>

        {
        util:recursiveRetrieving($i)
        }
    </item>
}
</display>
```

Figure 107 Script de formatage des données relatives au concept « interaction de forum » du modèle sémantique pour l'entrée de l'outil Tatiana

Moteur de gestion

Dans l'état actuel des choses, le moteur de gestion n'est pas encore développé. En effet, nous gérons actuellement les choses de manière expérimentale. Par exemple, nous n'avons pas encore développé un composant logiciel qui automatise l'exécution séquentielle d'un ensemble de scripts. Nous le faisons donc nous même manuellement. Actuellement, pour effectuer des opérations sur l'ontologie (ajout, édition, suppression) nous utilisons l'éditeur Protégé. Pour gérer (insérer, éditer, supprimer) les corpus et les scripts, nous le faisons directement dans eXist-db en utilisant le client d'administration livré avec la base de données.

Enfin, pour exécuter une requête XQuery sur le contenu des corpus, nous utilisons une interface offerte par eXist-db pour l’exécution de requêtes XQuery en utilisant le moteur XQuery intégré.

Réalisation : composants à développer

Moteur de gestion

Un travail de développement doit être consacré à ce composant essentiel pour améliorer l’utilisabilité de la plateforme. L'expérimentation présentée dans le chapitre 9 est en quelque sorte une « expérience de la pensée » : si le proxy était implémenté (ou plus exactement, le générateur de proxys, un module du moteur de gestion), que ferait-il ? La requête de l'analyse, basée sur les concepts du modèle sémantique et ceux attendus par l’outil d’analyse, induit les données à obtenir ; celles-ci sont à leur tour, après conversion, puis filtrage éventuel, à extraire du corpus. Ces besoins indiquent les scripts à utiliser (ou à programmer). Ce travail peut être automatisé : les différents scripts du système peuvent être repérés par une sémantique d'accès ou de conversion, et une signature de leurs arguments (sémantique et représentation des entrées et sorties). Le travail du proxy peut être relativement simple, en ce sens qu'il consiste en la création d'un assemblage de scripts d'extractions, suivi des conversions et formatages adéquats, pour obtenir l'opération compacte correspondant à l'extraction de données depuis un corpus C et la création d'une entrée pour l'outil O. Intuitivement, un outil opérant sur les signatures des scripts et un petit nombre de règles devrait permettre de réaliser cet assemblage. Un plus serait de créer l'assemblage sous la forme d'un outil graphique, tel que celui qui est décrit dans les perspectives. Cette présentation assurerait la documentation, et permettrait au chercheur de comprendre le travail du proxy. Le « script » obtenu, avec sa signature, pourrait être enregistré, et réutilisé. En outre, il pourrait être modifié, par exemple, pour changer certains paramètres, ou pour introduire un filtrage (comme dans la Figure 99, où l'on ne sélectionne que certains des items produits par le filtre). Plus généralement, le proxy se comporterait comme un conseiller pour l'utilisateur, en lui présentant les choix possibles, les scripts "voisins" de sa requête, et les scripts à développer.

Application Web

Nous avons présenté dans le chapitre 8 des exemples de scénarios d'utilisation de la plateforme « Beatcorp » et les avons illustrés par des maquettes d'interfaces Web pour l'exécution de ces scénarios. Le composant application Web accompagné du moteur de gestion améliorerait l'utilisabilité de plateforme de manière importante. En effet, pour manipuler l'ontologie, le chercheur n'aura qu'un seul outil lui permettant de manipuler les différents composants de la plateforme sans avoir besoin d'apprendre à manipuler eXist-db et protégé. Une telle application permettrait également de rendre transparents les traitements réalisés par le moteur de gestion. Cette application doit couvrir les fonctionnalités suivantes :

- gérer l'accès à la plateforme (public, privé) ;
- opérations : parcourir / insérer / supprimer / éditer / interroger sur la base de corpus et la base des scripts ;
- ajouter un outil d'analyse ;
- réaliser un outil d'analyse.

Manuel utilisateur

Nous présentons ici les parties à développer pour constituer un « manuel » décrivant l'implantation :

1. Description de l'approche « Proxyma ». Discussion sur l'intérêt de la conservation sans transformation des corpus et des résultats d'analyse. Description du « proxy », un outil permettant la création assistée des entrées pour les analyses ;
2. Les possibilités : ce que l'on peut faire avec ; quelques exemples tirés de la thèse ;
3. Les éléments du système :
    3.1. L'ontologie décrit les concepts pertinents aux corpus de traces d'interaction, aux analyses de ces traces, et aux objets concrets manipulés ;
    3.2. Les proxys, qui constituent un quasi substitut à l'informaticien qui auraient eu à rédiger les requêtes si une BDD traditionnelle avait été utilisée ;

3.3. Les scripts, qui représentent et implantent des fragments d'algorithmes, et peuvent être assemblés par les proxys pour fournir des accès aux données ;

4. Méthode d'intégration d'un nouveau corpus à la base, consistant à introduire le nouveau corpus, inchangé, dans la BDD, documenter le nouveau corpus et des ressources qui le composent, et lier les données du corpus aux concepts du modèle sémantique définie par l'ontologie ;

5. Méthode d'intégration d'un nouveau processus d'analyse (qui peut consister en un nouvel outil, ou une nouvelle utilisation d'un, ou plusieurs, outils existant). La méthode consiste à documenter les entrées attendues par le processus et les sorties de celui-ci, sous la forme de référence à des concepts, ou des agrégats de concepts de l'ontologie ;

6. Méthode d'analyse (sur des corpus et processus déjà intégrés) : construction ou adaptation d'un proxy existant.

Développement collaboratif

Le plein potentiel de l'outil apparaitrait si une communauté d'utilisateurs se constituait. La mise en place d'un site Web pour soutenir cette approche pourrait être ainsi décrite :

- Utiliser une approche de type "Wiki" pour une description partagée et collaborative de l'ontologie ;
- Placer les sources de « Beatcorp » sur un site permettant le développement collaboratif ;
- Construire, et maintenir, une base de scripts XQuery cohérents avec l'ontologie et les corpus existants ;
- Construire, et maintenir, une base d'outils d'analyse, et ajouter à l'ontologie les concepts nécessaires ;
- Gérer une liste des corpus rendus publics par les différentes équipes de recherche ;
- Mettre en place un forum de discussion pour tous ces aspects.

Conclusion

Le travail présenté dans ce manuscrit permet de notre point de vue de répondre aux verrous scientifiques identifiés dans le premier chapitre. L'approche « Proxyma » et l'architecture de la plateforme « Beatcorp » proposées permettent le partage de corpus hétérogènes de traces d'apprentissage contextualisées, et leur analyse avec des outils partagés. Le développement d'un prototype complet de la plateforme « Beatcorp » et basé sur l'approche « Proxyma » devrait permettre la validation de notre approche.

Bibliographie

Archives.org (2013). The Web archive, retrieved on December 2013 from : http://archive.org/web/web.php.

Avouris, N., Fiotakis, G., Kahrimanis, G., Margaritis, M., Komis, V. (2007). Beyond logging of fingertip actions : analysis of collaborative learning using multiple sources of data. *Journal of Interactive Learning Research, Special Issue: Usage Analysis in Learning Systems : Existing Approaches and Scientific Issues*, vol. 18(2). pp. 231-250.

Baker, M. J., Andriessen, J., Lund, K., van Amelsvoort, M., Quignard, M. (2007). Rainbow: a framework for analysing computer-mediated pedagogical debates. *International Journal of Computer-Supported Collaborative Learning*, *2*. pp. 315–357.

Baldauf, M., Dustdar, S., Rosenberg, F. (2007). A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 2, No. 4. pp. 263-277.

Baude, O., Eshkol, I. (2007). Entrer dans l'anonymat, Etude des «entités dénommantes»dans un corpus oral. *Colloque international Proper Names in Spoken Language*. Université de Bâle (Suisse).

Bolchini, C., Curino, A., C., Quintareli, E., Schreiber, A., F., Tanca, L. (2007). A Data-oriented Survey of Context Models. *SIGMOD Record*, Vol. 36, No. 4. pp. 19-26.

Bouhineau, D., Luengo, V., Mandran, N. (2013). Open platform to model and capture experimental data in Technology enhanced learning systems. *Alpine Rendez-Vous 2013.* Villars-de-Lans, Vercors, France.

Bouhineau, D., Luengo, V., Mandran, N. (2013). Share data treatment and analysis processes in Technology enhanced learning. *Alpine Rendez-Vous 2013.* Villars-de-Lans, Vercors, France.

Bratitsis, T., Dimitracopoulou, A. (2006). Indicators for Measuring Quality in Asynchronous Discussion Forae. In Proceedings of the IADIS Internartional Conference on Cognition and Exploratory Learning in Digital Age (CELDA 2006). Barcelona, Spain.

Butoianu, V., Vidal, P., Broisin, J. (2011). Prise en compte de la vie privée des usagers dans un Système à Base de Traces dédié à l'apprentissage en ligne. *Dans Environnement Informatique pour l'Apprentissage Humain (EIAH 2011)*. Mons, Belgique. pp. 355-367.

Butoianu, V., Vidal, P., Verbert, K., Duval, E., Broisin, J. (2010). User context and personalized learning: a federation of Contextualized Attention Metadata. *Journal of Universal Computer Science, John Wiley and Sons*, vol. 16, No. 16. pp. 2252-2271.

CALICO. (2010). The Calico Platform. Retrieved on July 2013 from : http://woops.crashdump.net/calicorss2/

Carron, T., Marty, J. C., France, L., Heraud, J. M. (2005). Preparing an Observed Pedagogical Experiment. *Cognition and Exploratory Learning in Digital Age, Kinshuk Sampson, Demetrios G. Isaias, Pedro T. (Ed.)*. Porto, Portugal. pp. 526-531.

Çelik, T. (2005). Attention.xml Technology Overview. Retrieved on July 2013 from http://tantek.com/presentations/2005/01/attentionxml.html

Champin, P. A., Prie, Y., Mille, A. (2004). MUSETTE: a framework for knowledge capture from experience. Dans 12ème Atelier du Raisonnement a Partir de Cas (RaPC'04). Villetaneuse, France. pp. 85-97.

Chanier, T., Ciekanski, M., Betbeder, M.L., Reffay, C., Lamy, M. N. (2010). Mulce : échanges de corpus d'apprentissage multimodaux (ANR-06-CORP-006). Accessible à : mulce-doc.univ-bpclermont.fr (consulté en Juillet 2013).

Chebil, H., Courtin, C., Girardot, J.-J. (2012). An Ontology-Based Approach for Sharing and Analyzing Learning Trace Corpora. *In proceedings of the IEEE Sixth International Conference on Semantic Computing ICSC 2012*. Palermo, Italy. pp. 101-108.

Chebil, H., Courtin, C., Girardot, J.-J. (2012). The Proxy Model: A New Approach to Sharing and Analyzing Learning Traces Corpora. *International Journal of Information and Education Technology*. Vol. 2, N° 4. pp. 208-211.

Chebil, H., Courtin, C., Girardot, J.-J. (2011). BEATCORP : une plateforme de benchmarking pour l'analyse de corpus de traces. *Workshop "Partager des données d'observation pour la recherche en EIAH - traces d'activité d'apprentissage" EIAH 2011*. Mons, Belgique.

Choquet, C. (2007). *Ingénierie et réingénierie des EIAH - L'approche REDiM.* Thèse d'Habilitation à Diriger des Recherche en Informatique. Laboratoire d'Informatique de l'Université du Maine (LIUM).

Choquet, C., Iksal, S. (2007). Modeling Tracks for the Model Driven Re-engineering of a TEL System. *Journal of Interactive Learning Research*, vol. 18 No. 2. pp. 161-184.

CIM. (2013). Common Information Model, retrieved on July 2013 from : http://dmtf.org/standards/cim

Corbel, A., Girardot, J.J., Lund, K. (2006). A Method for Capitalizing upon and Synthesizing Analyses of Human Interactions. *In proceedings of the EC-TEL06 Workshops. Crete, Greece*.

Corbel, A., Jaillon, P., Serpaggi, X., Baker, M., Quignard, M., Lund, K. (2003). DREW : Un outil Internet pour créer des situations d'apprentissage coopérants. *Dans Desmoulins, Marquet, & Bouhineau, Actes de la conférence EIAH 2003*. Strasbourg, France. pp. 109-113.

Courtin C. (2008). CARTE: an Observation Station to Regulate Activity in a Learning Context. *In proceedings of the Fifth IADID International Conference Cognition and Exploratory Learning in Digital Age*. Freiburg, Germany. pp. 191-197.

Courtin C., Talbot S. (2007). Une Station d'Observation pour des Situations d'Apprentissage Collaboratif Instrumenté. *Actes INRP / EIAH 2007*. Lausanne, Suisse. pp. 371-376.

CTAT. (2003). Cognitive Tutor Authoring Tools, retrieved on July 2013 from: http://ctat.pact.cs.cmu.edu/

DCMI. (1994). Dublin Core Metadata Initiative. Project website retrieved on July 2013: http://dublincore.org/

De Chiara, R., Di Matteo, A., Manno, I., Scarano, V. (2007). CoFFEE: Cooperative face2face educational environment. *In proceedings of the third Interanational Conference on Collaborative Computing: Networking, Applications and Worksharing*. New York, USA. pp. 243-252.

De Wever, B., Schellens, T., Valcke, M., Van Keer, H. (2006). Content analysis schemes to analyse transcripts of online asynchronous discussion groups: a review. *Computers & Education.* Vol. 46, No. 1. pp 6-28.

Djouad, T., Settouti, L.S., Mille, A., Reffay, C., Prié, Y. (2010). SBT-IM : Un Système à Base de Traces pour le calcul des indicateurs d'interaction dans Moodle. *Rapport de recherche RR-LIRIS-2010-002*.

Djouad, T. (2008). Analyser l'activité d'apprentissage collaboratif : Une approche par transformations spécialisées de traces d'interactions. *Dans Secondes Rencontres Jeunes Chercheurs en EIAH, RJC-EIAH'2008*. Lille, France. pp. 93-98.

Di Eugenio, B., Fossati, D., Yu, D., Haller, S. (2005). Aggregation improves learning: experiments in natural language generation for intelligent tutoring systems. *In ACL05, Proceedings of the 42nd Meeting of the Association for Computational Linguistics*.

Drachsler, H., Bogers, T., Vuorikari, R., Verbert, K., Duval, E., Manouselis, N., Beham, G., Lindstaedt, S., Stern, H., Friedrich, M., Wolpers, M. (2010). Issues and considerations regarding sharable data sets for recommender systems in technology enhanced learning. *In N. Manouselis, H. Drachsler, K. Verbert, O. Santos, eds, Proceedings of Elsevier Procedia Computer Science.* Vol. 1. pp. 2849-2858.

DTD. (2008). Document Type Definition. Retrieved on July 2013 from : http://www.w3.org/TR/REC-xml/#dt-doctype

Duclosson N. (2004). Représentation des connaissances dans l'EIAH AMBRE-add. *Technologies de l'Information et de la Connaissance dans l'Enseignement supérieur et l'industrie, TICE'2004*. Compiègne, France. pp. 164-171.

Dyke, G. (2009). *Un modèle pour la gestion et la capitalisation d'analyses de traces d'activités en interaction collaborative*. Thèse de doctorat en Informatique. Ecole Nationale Supérieure des Mines de Saint-Etienne.

Dyke, G., Lund, K., Girardot, J. J. (2009). Tatiana: an environment to support the CSCL analysis process. *In proceedings of the International Conference on Computer Supported Collaborative Learning (CSCL 2009)*. Rhodes, Greece. pp. 58-67.

Dyke, G., Lund, K., Girardot, J.J. (2008). TATIANA: Un logiciel pour l'analyse des interactions humaines médiatisées par ordinateur, de la spécification à l'implémentation. *Rapport de recherche, 2008-400-005*, Centre G2I-EMSE / Laboratoire CNRS/ICAR.

Eclipse. (2013). Eclipse Integrated Development Environment. Website retrieved on July 2013 from : http://www.eclipse.org/

Euzenat, J., Le Bach, T., Barrasa, J., Bouquet, P., De Bo, J., Dieng-Kuntz, R. , Ehrig, M., Hauswirth, M., Jarrar, M., Lara, R., Maynard, D., Napoli, A., Stamou, G., Stuckenschmidt, H., Shvaiko, P., Tessaris, S., Van Acker, S., Zaihrayeu, I. (2004). D2.2.3: State of the art on ontology alignment. *Knowledge Web project, realizing the semantic web.*

eXist-db. (2012). eXist-db Open Source Native XML Database. Website retrieved on July 2013 from : http://exist-db.org/exist/apps/homepage/index.html

Ferraris, C., Lejeune, A. (2009). Analyse de traces et personnalisation des environnements informatiques pour l'apprentissage humain, Chapitre 5, Hermes/Lavoisier, 2009. pp. 219-244.

George, S. (2003). Forum contextuel : une étude pour le téléenseignement. *Dans actes de la 15ème Conférence Francophone sur l'Interaction Homme-Machine (IHM 2003)*. Caen, France. pp. 104-111.

Georgeon, O., Henning, M., Bellet, T., Mille, A. (2007). Creating cognitive models from activity analysis: a knowledge engineering approach to car driver modeling. *In proceedings of the International Conference on Cognitive Modeling*. Ann Arbor, Michigan. pp. 43-48.

Gertner, A. et VanLehn, K. (2000). Andes: A Coached Problem Solving Environment for Physics. *In proceedings of then Intelligent Tutoring Systems: 5th International Conference*. Berlin: Springer (Lecture Notes in Computer Science, Vol. 1839), pp. 133-142.

Giguet, E., Lucas, N., Blondel, F. M., Bruillard, E. (2009). Share and explore discussion forum objects on the Calico website. *8th internation conference on Computer Supported Collaborative Learning (CSCL 2009)*. Rhodes, Grèce. pp. 616-620.

Granitzer, M., Rath, A. S., Kröll, M., Seifert, C., Ipsmiller, D., Devaurs, D., Weber, N., Lindstaedt, S. (2009). Machine learning based work task classification. *Journal of Digital Information Management*, vol. 7, No. 5. pp. 306-314.

Harrer, A., Zeini, S., Kahrimanis, G., Avouris, N., Marcos, J.A., Martinez-Mones, A., Meier, A., Rummel, N., Spada, H. (2007). Towards a Flexible Model for Computer-based Analysis and Visualisation of Collaborative Learning Activities. *In Proceedings of CSCL 2007*. New Jersey, USA.

IEEE 1484.12.3. (2003). Standard for XML binding for Learning Object Metadata data model.

Iksal S., Choquet C. (2005). An Open Architecture for Usage Analysis in a E-Learning Context. *In Proceedings of the 5th IEEE International Conference on Advanced Learning Technologies (ICALT'2005)*. Taïwan, China. pp. 177-181.

IMS-CP. (2009). IMS Content Packaging Specification. http://www.imsglobal.org/content/packaging/cpv1p1p4/imscp_bestv1p1p4.html .

IMS GLC. (1997). IMS Global Learning Consortium. Website : http://www.imsglobal.org/

IMS-LD. (2003). IMS Learning Design Specification Version 1, final specification. Retrieved from http://www.imsglobal.org/learningdesign/

Jermann, P., Soller, A., Mühlenbrock, M. (2001). From mirroring to guiding: A review of state of the art technology for supporting collaborative learning. *In Proceedings of the First European Conference on Computer-Supported Collaborative Learning*. Maastricht, the Netherlands. pp. 324–331.

Kalfoglou, Y., Schorlemmer, M. (2003). Ontology mapping: the state of the art. *The Knowledge Engineering Review*. Vol. 18(1). pp 1–31.

King, G. (2007). An Introduction to the Dataverse Network as an Infrastructure for Data Sharing." Sociological Methods and Research 36(2):173–199. http://gking.harvard.edu/files/abs/dvn-abs.shtml (2007)

Kirschenmann, U., Scheffel, M., Wolpers, M. (2010). An Attempt to Close the Gap: Recommending Learning Activities in PLE. *In Proceedings of the 13th International Conference on Interactive Computer aided Learning (ICL 2010)*. September 2010. Hasselt, Belgium

Koedinger, K.R., Cunningham, K., Skogsholm, A., Leber, B. (2008). An open repository and analysis tools for fine-grained, longitudinal learner data. *In Proceedings of the First International Conference on Educational Data Mining*. Montréal, Québec. pp. 157-166.

Laflaquière J., Prié Y. (2008). Ingénierie des traces numériques d'interaction comme inscriptions de connaissances. *Actes de la conférence Ingénierie des Connaissances*. Nancy, France.

Lalle, S., Luengo, V., Guin, N. (2013). Assistance à la conception de techniques de diagnostic des connaissances. *Dans actes de la conférence EIAH 2013*. Toulouse, France.

Lejeune A., Pernin, J.P. (2004). A Taxonomy for Scenario-based Engineering. *Cognition and Exploratory Learning in Digital Age* (CELDA 2004). Lisbonne, Portugal. pp.249-256.

LOM. (2002). IEEE-SA Standard 1484.12.1-2002. Final draft retrieved on July 2013 from : http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf

Luengo V., Vadcard L., Dubois M., Mufti-Alchawafa D. (2006). TELEOS : de l'analyse de l'activité professionnelle à la formalisation des connaissances pour un environnement d'apprentissage. *Actes de la conférence « Ingénierie de Connaissances », IC 2006*. Nantes, France.

Lund, K. Mille, A. (2009). Traces, Traces d'interactions, Traces D'apprentissages Définitions, Modèles Informatiques, Structurations, Traitements et Usages (Eds. J.C. Marty & A. Mille). *Analyse de traces et Personnalisation des EIAH dans la collection Traité Informatique et Systèmes d'Information*. Lavoisier-Hermes : Paris. pp. 21-56.

Martel, C., Vignollet, L., Ferraris, C., David, J. P., Lejeune, A. (2006). Modeling collaborative learning activities on e-learning platforms. *In proceedings of the IEEE*

International Conference on Advanced Learning Technologies (*ICALT 2006)*. Kerkrade, Netherlands.

Martínez, A., Harrer, A., Barros, B. (2005). Library of interaction analysis methods. *Delivrable of the ICALTS JEIRP*.

Martínez, A., de la Fuente, P., Dimitriadis, Y. (2003). Towards an XML-based representation of collaborative interactions. *In proceedings of the International Conference on Computer Supported Collaborative Learning (CSCL 2003)*, Bergen, Norway. pp. 379–384.

May, M. (2009). *Using tracking data as reflexive tools to support tutors and learners in distance learning situations: an application to Computer-Mediated Communications.* Thèse de doctorat en Informatique. Institut National des Sciences Appliquées de Lyon.

May, M., George, S., Prévôt, P. (2008). A closer look at tracking human and computer interactions in web-based communications. *International Journal of Interactive Technology and Smart Education (ITSE)*, Vol. 5, No. 3. pp 170-188.

Moodle. (2013). Learning Management System, retrieved on July 2013 from : https://moodle.org/

Mostow, J., Aist, G. (2001). Evaluating tutors that listen: An overview of Project LISTEN, in *Smart Machines in Education: The coming revolution in educational technology*., K. Forbus and P. Feltovich, ed., MIT/AAAI Press, 2001, pp. 169 - 234.

MULCE. (2013). Mulce Learning and Teaching Corpora repository. Retrieved on July 2013 from : http://lrl-diffusion.univ-bpclermont.fr/mulce2/index.html

MULE-Doc. (2013). Mulce.org documentation. Retrieved on July 2013 from : http://lrl-diffusion.univ-bpclermont.fr/mulce2/index.html

Muñoz-Merino, P.J., Wolpers, M., Delgado Kloos, C., Muñoz-Organero, M., Friedrich, M. (2010). An Approach for the Personalization of Exercises based on Contextualized Attention Metadata and Semantic Web technologies. *In Proceedings of the 10th IEEE International Conference on Advanced Learning Technologies (ICALT 2010).* Sousse, Tunisia, July 2010.

Nicaud, J. F., Bouhineau, D., Huguet, T. (2002). The Aplusix-Editor: A New Kind of Software for the Learning of Algebra. *Proceedings of the 6th International Conference on Intelligent Tutoring Systems, LNCS 2363*. Biarritz, France and San Sebastian, Spain, p 178-189.

Noras, M., Reffay, C., Betbeder, M. L. (2007). Structuration de corpus de formation en ligne en vue de leur échange. *Actes EIAH 2007 : Environnements Informatiques pour l'Apprentissage Humain*. Lausanne, Suisse. pp. 59-64.

OAI-PMH. (2013). Open Archives Initiative Protocol for Metadata Harvesting, retrieved on November 2013 from : http://www.openarchives.org/pmh/

OWL. (2004). Ontology Web Language, retrieved on July 2013 from : http://www.w3.org/2004/OWL/

PSLC Datashop. (2013). PSLC Datashop, a data analysis service for the learning science community. Retrieved on July 2013 from : https://pslcdatashop.web.cmu.edu/

Psotka, J., Massey, D. L., Mutter, S. A., Brown, J.S. (1988). *Intelligent Tutoring Systems: Lessons Learned*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Rath, S. A., Devaurs, D., Lindstaedt, N. S. (2009). UICO: An Ontology-Based User Interaction Context Model for Automatic ask Detection on the Computer Desktop, In Proceedings of the 1st Workshop on Context, Information and Ontologies. Heraklion, Greece.

RDF. (2004). Resource Description Framework. Retrieved on July 2013 from : http://www.w3.org/RDF/

RDFS. (2004). RDF Vocabulary Description Language (RDF Schema). Retrieved on July 2013 from : http://www.w3.org/TR/rdf-schema/

Reffay, C., Blondel, F.M., Giguet, E. (2012). Stratégies pour l'anonymisation systématique d'un corpus d'interactions plurilingues. Degache, C. & Garbarino, S. (Ed.) (2012). Actes du colloque IC2012. *Intercompréhension : compétences plurielles, corpus, intégration*. Université Stendhal Grenoble 3 (France), 21-22-23 juin 2012.

Reffay, C., Betbeder, M.L. (2009). Sharing corpora and tools to improve interaction analysis. In Proceedings of the Fourth European Conference on Technology Enhanced Learning "Learning in the Synergy of Multiple Disciplines". Nice, France. pp 196-210.

Reffay, C., Chanier, T., Noras, M., Betbeder, M.L. (2008). Contribution à la structuration de corpus d'apprentissage pour un meilleur partage en recherche. *Revue Sticef* (Sciences et Technologies de l'Information et de la Communication pour l'Education et la Formation), numéro spécial EPAL (échanger pour apprendre en ligne), Vol.15, http://sticef.org.

Reffay C, Teutsch P. (2007). Anonymisation de corpus réutilisables. *Actes de EIAH'2007*, Lausanne, Suisse.

RFC-5646. (2009). Tags for identifying languages. Retrieved on July 2013 from : http://tools.ietf.org/html/rfc5646

RSS. (2009). Really Simple Syndication. Retrieved from : http://www.rssboard.org/rss-specification.

Scheffel, M., Beer, F., Wolpers, M. (2010). Analysing Contextualized Attention Metadata for Self-regulated Learning. *In Proceedings of the 2nd International Conference on Computer Supported Education*. Valencia, Spain, April 2010

Settouti, L., Guin, N., Luengo, V., Mille, A. (2011). Adaptable and resusable query patterns for trace-based learner modeling. *In proceedings of EC-TEL, Sixth European Conference on Technology Enhanced Learning : Towards Ubiquitous Learning*. Palermoo, Italy. September 2011. pp. 384-397.

Settouti, L., Guin, N., Mille, A., Luengo, V. (2010). A trace-based learner modeling framework for Technology-Enhanced Learning systems. *In proceedings of the 10th*

IEEE International Conference on Advanced Learning Technologies, Sousse, Tunisie. pp. 73-77.

Settouti, L.S. (2011). Systèmes à base traces modélisées : modèles et langages pour l'exploitation des traces d'interaction. Thèse de doctorat en informatique. Université Claude Bernard Lyon 1, Lyon.

Settouti, L.S., Prié, Y., Marty, J.C., Mille, A. (2009). A trace-based system for technology-enhanced learning systems personalization. *In proceedings of the 9th IEEE International Conference on Advanced Learning Technologies*. Riga. July 2009. pp. 93-97.

Settouti, L.S., Prié, Y., Marty, J.C., Mille, A. (2007). Vers des Systèmes à Base de Traces modélisées pour les EIAH. *Rapport de recherche RR-LIRIS-2007-016*.

SPARQL. (2008). SPARQL Query Language for RDF. Retrieved on July 2013 from : http://www.w3.org/TR/ rdf-sparql-query/

Strang, T., Linnhoff-Popien, C. A context modeling survey. (2004). *In Workshop on Advanced Context Modelling, Reasoning and Management*, UbiComp '04.

Talbot, S., Courtin, C. (2008). Trace Analysis in Instrumented Learning Groupware: an experiment in a practical class at the university. *In Proceedings of the Seventh IASTED International Conference WEB-BASED EDUCATION 2008*, Innsbruck, Austria, March 17-19 2008. pp. 418-422.

Tao, F., Millard, D., Zalfan, M., Chen, L., Davis, H. (2007). Knowledge based Learning Experience Management on the Semantic Web. *In: IADIS International Conference of e-Learning*. Lisbon, Portugal.

Tatiana. (2009). Trace Analysis Tool for Interaction Analysts. Downloadable on : http://code.google.com/p/tatiana/ (retrieved on July 2013)

The Dataverse Network, project webpage retrieved on November 2013 from : http://thedata.org/

Tsarkov, D., Horrocks, I. (2006). FaCT++ description logic reasoner: System description. *In proceedings of the International Joint Conference on Automated Reasoning (IJCAR 2006).* Seattle, WA, USA.

Tutor Message Format (2013). Guide to the Tutor Message Format, retrieved on July 2013 from : http://pslcdatashop.org/dtd/guide/

Wolpers, M., Najjar, J., Verbert, K., Duval, E. (2007). Tracking Actual Usage: the Attention Metadata Approach. *International Journal Eeducational Technology and Society*. 10 (3). pp. 106-121.

Wolpers, M., Grohmann, G. (2005). PROLEARN: technology-enhanced learning and knowledge distribution for the corporate world. *In International Journal on Knowledge and Learning, 1(1/2).* pp. 44-61.

XML. (2008). eXtensible Markup Language. Version 1.0 (fifth edition). Retrieved on July 2013 from : http://www.w3.org/XML/

XPath. (1999). XML Path Language. Retrieved on July 2013 from : http://www.w3.org/TR/xpath/

XQuery. (2010). XQuery 1.0 : An XML Query Language (second edition). Retrieved on July 2013 from : http://www.w3.org/TR/xquery/

XSD. (2004). XML Schema. Retrieved on July 2013 from : http://www.w3.org/XML/Schema

Publications

Chebil, H., Courtin, C., Girardot, J.-J. (2012). An Ontology-Based Approach for Sharing and Analyzing Learning Trace Corpora. *In proceedings of the IEEE Sixth International Conference on Semantic Computing ICSC 2012.* Palermo, Italy. pp. 101-108.

Chebil, H., Courtin, C., Girardot, J.-J. (2012). The Proxy Model: A New Approach to Sharing and Analyzing Learning Traces Corpora. *International Journal of Information and Education Technology*. Vol. 2, N° 4. pp. 208-211.

Chebil, H. (2012). Corpus de traces d'activité dans les EIAH : modélisation, étude d'une plate-forme de gestion et application à la construction de corpus de référence. Poster présenté à la Journée de la recherche EDSIS 12 juin 2012.

Chebil, H., Courtin, C., Girardot, J.-J. (2011). BEATCORP : une plateforme de benchmarking pour l'analyse de corpus de traces. *Workshop "Partager des données d'observation pour la recherche en EIAH - traces d'activité d'apprentissage" EIAH 2011*. Mons, Belgique.

Chebil, H. (2010). Corpus de traces d'activité dans les EIAH : modélisation, étude d'une plate-forme de gestion et application à la construction de corpus de référence. Poster présenté aux Journées scientifiques du cluster ISLE (01 et 02 Décembre 2010).

Chebil, H. (2010) *Réflexions autour de la création d'un corpus de traces d'interactions au sein du projet ISLE/EIAH. Dans actes des Troisièmes Rencontres Jeunes Chercheurs en EIAH, RJC-EIAH'2010,* Lyon, France.Mai 2010.

École Nationale Supérieure des Mines
de Saint-Étienne

NNT : *Communiqué le jour de la soutenance*

Hajer CHEBIL

ACTIVITY TRACE CORPORA IN TECHNOLOGY ENHANCED LEARNING ENVIRONMENTS : MODELING, ANALYSIS AND MANAGEMENT PLATFORM, APPLICATION FOR THE CONSTRUCTION OF REFERENCE CORPORA

Speciality : Computer Science

Keywords : trace corpus, activity trace, learning, analysis, data and analysis tools sharing

Abstract :

This work is part of the "TEL Environments customization" project studying the use of activity traces in the evaluation and customization of mediated learning situations. Analyzing traces is generally performed using analysis tools developed by researchers specifically for their needs. Published research results can generally not be verified or compared, due to difficulties of sharing corpora and analysis tools. The aim of this research work is to provide researchers using TEL environments in their researches with a platform to share corpora of contextualized interaction traces and analyses performed on them, and to analyse those corpora using shared analysis and visualization tools. Heterogeneity of traces produced by TEL environments, due to the diversity of learning domains and to analysis needs makes the proposition of a common representation cannot satisfy the various needs of multidisciplinary researchers. We propose the "proxy" approach, a participative and incremental solution based on an ontology which defines three models: a corpus model defining the structure and description metadata of the corpus and its contents, a semantic model defining generic concepts which can be retrieved in shared corpora, and an operational model defining a set of operations ensuring interoperability between shared corpora and analysis tools. Based on this approach, we propose a platform architecture for sharing traces corpora and analysis tools allowing researchers to share their own corpora, to access to shared corpora, and to analyze them.

École Nationale Supérieure des Mines
de Saint-Étienne

NNT : *Communiqué le jour de la soutenance*

Hajer CHEBIL

CORPUS DE TRACES D'ACTIVITÉ DANS LES ENVIRONNEMENTS INFORMATIQUES POUR L'APPRENTISSAGE HUMAIN : MODÉLISATION, ÉTUDE D'UNE PLATEFORME DE GESTION ET APPLICATION À LA CONSTRUCTION DE CORPUS DE RÉFÉRENCE

Spécialité: Informatique

Mots clefs : corpus de traces, traces d'activité, apprentissage, analyse, partage de données et d'outils d'analyse

Résumé :

Ce travail s'inscrit dans le projet « Personnalisation des EIAH » étudiant l'utilisation des traces d'activité dans l'évaluation et la personnalisation des situations d'apprentissage médiatisées. L'analyse des traces se fait généralement par des outils d'analyse réalisés par les chercheurs et spécifiques à leurs besoins. Les résultats publiés ne peuvent généralement pas être vérifiés ni comparés, ceci étant dû aux difficultés de partage des corpus et des outils d'analyse. L'objectif de ce travail est de proposer aux chercheurs utilisant les EIAH, une plateforme pour le partage, d'une part des corpus de traces d'interaction contextualisées et les analyses réalisées sur ces corpus, et des outils d'analyse et de visualisation des traces d'autre part. L'hétérogénéité des traces produites par les EIAH, due à la variété des domaines d'apprentissage et des besoins d'analyse, fait que la proposition d'une représentation commune ne peut répondre aux différents besoins de chercheurs pluridisciplinaires. Nous proposons l'approche par « proxy », une solution participative et incrémentale basée sur une ontologie qui définit trois modèles : un modèle de corpus définissant la structure et les métadonnées de description du corpus et son contenu, un modèle définissant les concepts génériques pouvant être retrouvés dans les corpus, et un modèle opérationnel définissant des opérations assurant l'interopérabilité entre un corpus et un outil d'analyse partagé. En nous basant sur cette approche, nous proposons une architecture de plateforme de partage de corpus de traces et d'outils d'analyse permettant aux chercheurs de partager leurs corpus, d'accéder aux corpus partagés, et de les analyser.