



HAL
open science

Interprétation collaborative de séries temporelles. Application à des données de réanimation médicale.

Thomas Guyet

► **To cite this version:**

Thomas Guyet. Interprétation collaborative de séries temporelles. Application à des données de réanimation médicale.. Informatique [cs]. Institut National Polytechnique de Grenoble - INPG, 2007. Français. NNT: . tel-00264145

HAL Id: tel-00264145

<https://theses.hal.science/tel-00264145v1>

Submitted on 14 Mar 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

N° attribué par la bibliothèque

□□□□□□□□□□

T H E S E

pour obtenir le grade de

DOCTEUR DE L'INP Grenoble

Spécialité : *Interactions, connaissance et inférence*

préparée au laboratoire TIMC (Techniques de l'Ingénierie Médicale et de la Complexité)
dans le cadre de **l'Ecole Doctorale** *Mathématiques, Sciences et Technologie de l'Information*

présentée et soutenue publiquement

par

Thomas GUYET

le 11 décembre 2007

**INTERPRÉTATION COLLABORATIVE DE SÉRIES TEMPORELLES MULTIVARIÉES.
APPLICATION À DES DONNÉES DE RÉANIMATION MÉDICALE.**

Directrice de Thèse : Mme Catherine GARBAY
Co-Directeur de Thèse : M. Michel DOJAT

JURY

M. Christian Jutten	Professeur de l'INPG, LIS	<i>Président</i>
M. Guy Carrault	Professeur de l'Université de Rennes 1, LTSI	<i>Rapporteur</i>
M. François Pachet	Chercheur HdR, Sony CSL-Paris	<i>Rapporteur</i>
Mme Catherine Garbay	Directrice de Recherche CNRS, LIG	<i>Directrice de thèse</i>
M. Michel Dojat	Ingénieur de Recherche Inserm, Institut des Neurosciences	<i>Co-Directeur de thèse</i>
M. Pierre Baconnier	Professeur de l'UJF, Praticien Hospitalier du CHU Grenoble	<i>Examineur</i>

Remerciements

Je remercie Guy Carrault et François Pachet d'avoir accepté d'être les rapporteurs de cette thèse, Christian Jutten d'avoir accepté de présider ce jury et P. Baconnier d'avoir accepté d'évaluer ce travail.

Je remercie chaleureusement mes directeurs de thèse, Catherine Garbay et Michel Dojat. Après 4 ans passés sous leur direction, je continue à qualifier leur encadrement de complémentaire, attentif, efficace mais compréhensif et ce avec une bonne ambiance dans toutes les situations. Beaucoup de choses contenues dans ce manuscrit n'auraient jamais été couchées sans leurs incitations et leur science de la maïeutique.

Ce travail a été également soutenu par deux équipes de recherche : MAGMA/LIG et PRETA/TIMC. J'en remercie donc les chefs respectifs Y. Demazeau et P. Baconnier pour m'y avoir accueillis, les personnels de recherche avec qui j'ai pu échanger sur mon travail ainsi que les personnels administratifs et techniques pour leur accompagnement. Je remercie également l'équipe du professeur L. Brochard et en particulier le docteur A. Thille pour les données médicales fournies et le temps qu'ils m'ont consacré.

Plus personnellement, j'adresse également des remerciements amicaux : aux membres du TIMC que j'ai côtoyés et appréciés pour échanger sur des sujets loin de mes problématiques professionnelles et en particulier l'équipe RFMQ (Julie, Françoise, Marie-Paule, Malika, ...) et Bruno sans qui j'aurai certainement bu moins de café, mes colocataires de bureau pour m'avoir supporté au jour le jour, mes comparses de l'atelier de monitoring ainsi que mes collègues de l'IUT Chambéry. Je remercie également très amicalement Sebastien pour son soutien pendant ces années.

Et enfin, je remercie mes amis et mes parents pour l'évidence de leur soutien qui est précieuse ...

Table des matières

Introduction	3
1 Amélioration du monitoring en unité de soins intensifs	7
A Monitoring de patients en Unités de Soins Intensifs (USI)	8
B Amélioration des systèmes de monitoring de patients en USI	16
C Construction de modèles pour le monitoring	22
D Des systèmes pour aider à l'exploration de séries temporelles	25
2 Interprétation collaborative de séries temporelles	35
A Interprétation	36
B Interprétation des séries temporelles	47
C Collaboration Homme/Machine	52
D Interprétation collaborative	63
3 Un système de support à l'interprétation collaborative de séries temporelles	71
A Architecture générale : Un système multi-agents	72
B Modèles	83
C Principaux algorithmes de traitements des données	92
4 Évaluations	107
A Méthodologie d'évaluation de l'interprétation collaborative	108
B Évaluations de la faisabilité de la collaboration	117
C Évaluation de la pertinence de la collaboration	139
D Utilisation pour l'interprétation de séries temporelles de patients ventilés	145
5 Discussions et perspectives	151
A L'Enaction et l'annotation comme approche de l'interprétation collaborative	152
B Réalisation d'un système sur les principes de l'Enaction	155
C Réponse de l'approche au problème applicatif	159
Conclusion	163
Bibliographie	164
Annexes	174
A Publications	177
B Algorithme MNClassification	178
C Algorithme d'apprentissage de motifs temporels	183
Table des matières détaillée	187
Table des figures et Liste des tableaux	190

Introduction

Cette thèse propose une approche de la collaboration homme-machine dans le domaine de l'interprétation de données complexes qui se fonde sur les principes d'autonomie et d'adaptation des partenaires. Son objet est donc la spécification des compétences que doit posséder un tel système, et des modalités qui lui permettent d'interagir avec son partenaire humain.

Le domaine d'application de cette thèse est la problématique difficile de l'extraction de modèles à partir de séries temporelles provenant d'enregistrements de patients en unité de soins intensifs (USI), nous mettons en place une telle situation collaborative et nous proposons un système capable d'assurer cette collaboration.

Les USI sont des unités de soins dans lesquelles les patients nécessitent une surveillance en continu. Pour cela, le personnel médical s'aide d'une multitude d'outils de surveillance automatique. Les systèmes de monitoring sont utilisés en USI pour assurer la surveillance de paramètres physiologiques et l'alerte au personnel soignant lorsqu'une situation est supposée critique par la machine. Malheureusement, ces systèmes sont générateurs d'un grand nombre de fausses alarmes et doivent donc être améliorés pour permettre d'assurer des soins de meilleure qualité. Une des voies possibles pour l'amélioration de ces systèmes est l'utilisation de la reconnaissance dans les signaux physiologiques et le raisonnement à partir d'évènements et de scénarios (suites caractéristiques d'évènements).

Cependant de tels évènements ou scénarios n'existent pas sous une forme utilisable par un système automatique. Il est donc nécessaire de formaliser cette connaissance. Une première approche aurait pu consister à demander aux cliniciens de formaliser ces connaissances en leur demandant de rédiger de manière formelle des scénarios typiques qu'ils peuvent observer. Mais un clinicien n'a généralement qu'une connaissance pratique de ces scènes typiques. La tâche de formalisation s'annonce donc difficile et longue. Une seconde approche pourrait nous conduire à utiliser des données d'enregistrements de patients pour identifier dans les données les scénarios dont on a besoin. Sans la connaissance du clinicien, cette exploration serait vouée à l'échec car trop difficile sans l'introduction de connaissances qui permettraient d'orienter les traitements. De plus, elle pourrait aboutir à un résultat *in fine* sans intérêt car sans relation avec la réalité pratique des données qui sont traitées.

Notre approche consiste donc à proposer une solution mixte, *i.e.* un système informatique d'aide à la construction de scénarios à partir de séries temporelles. Et plutôt que de faire réaliser explicitement la tâche de la construction des scénarios, nous faisons réaliser une interprétation de séries temporelles (les enregistrements de patients) qui implique la construction des modèles d'évènements et de scénarios.

Nous avons souhaité explorer une solution dans laquelle le système serait plutôt un collaborateur pour le système que simplement un outil calculatoire répondant à des requêtes de son utilisateur. En effet, nous pensons que dans un problème tel que l'interprétation de signaux le *point de vue* sur les données est déterminant sur le résultat qui peut être obtenu. Ainsi, l'idée de coupler un utilisateur à un système

informatique doit se faire dans l'objectif de pouvoir profiter de deux points de vues différents sur les données qui peuvent apparaître comme complémentaires. Le clinicien a un point de vue des données par une connaissance de la physiologie et une approche globale d'un patient particulier, tandis que la machine aura un point de vue informationnel sur les données et s'appuiera sur ses capacités de calculs pour traiter un grand ensemble des patients.

Nous nous sommes appuyés sur le paradigme de l'Enaction de F. Varela et H. Maturana pour concevoir la situation de collaboration entre l'homme et la machine. Ces paradigmes nous ont amenés à penser la collaboration entre le système et la machine comme un couplage structurel dans lequel les deux agents construisent de manière autonome leur propre interprétation et où les interactions mutuelles les influencent structurellement. C'est-à-dire qu'au cours des interprétations successives, un agent doit apprendre et donc s'adapter pour tenir compte des interactions de l'autre agent.

L'autonomie entre les deux agents doit être respectée pour assurer la préservation de la spécificité des points de vue sur les données pendant l'interprétation. Nous proposons de faire interagir les agents au moyen d'annotations dont la sémantique leur est commune. Les annotations qui peuvent être réalisées sur les données sont des segments, permettant de pointer des parties intéressantes des séries, des symboles pour associer les parties de séries avec des événements et des scènes pour associer des parties des enregistrements avec des scénarios.

Il est facile d'envisager que le clinicien puisse apprendre des annotations qui sont réalisées par la machine. En revanche, notre approche n'est viable que si nous sommes capable de proposer un système informatique qui puisse définir sa propre interprétation des données, de la transmettre au moyen d'annotations et également de s'adapter aux annotations de l'utilisateur. Nous avons donc développé un système doté de ces capacités.

Nous avons réalisé un système capable d'interpréter des séries temporelles, en particulier de construire des modèles d'évènements et des scénarios, et de collaborer avec un clinicien au moyen d'annotations. Ce système est conçu comme un système multi-agents qui construit une interprétation d'une série temporelle en suivant le schéma de la figure 1.

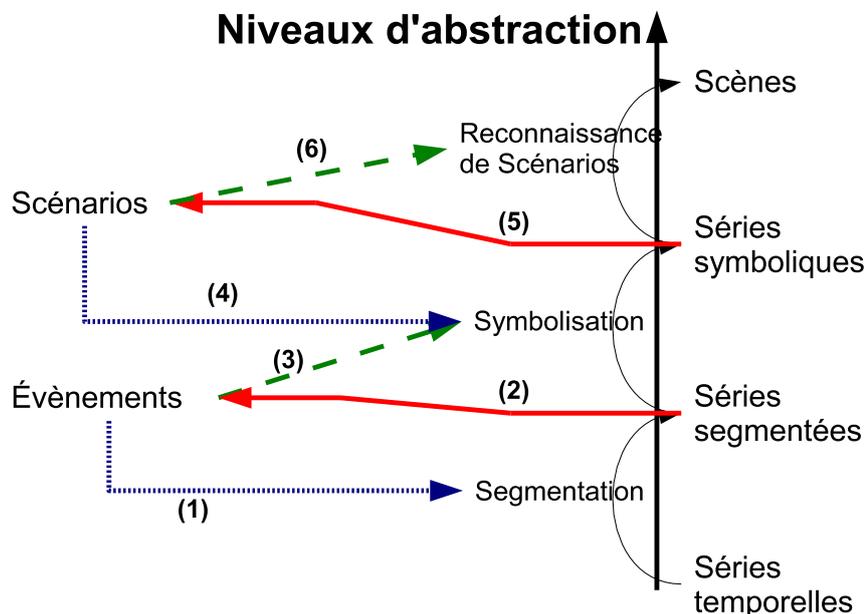


FIG. 1. Schéma général de l'interprétation des séries temporelles.

Cette thèse est organisée comme suit : Dans un **premier chapitre**, nous présentons le contexte général du monitoring de systèmes dynamiques et du monitoring de patient en unité de soins intensifs. Dans le **second chapitre**, nous explorons la notion de l'interprétation et de la collaboration afin de proposer une approche originale inspirée de la paradigme de l'Enaction. Notre approche est centrée

sur une interprétation collaborative de séries temporelles réalisée par un clinicien et une machine chacun considéré comme un agent (autonome) embarqué dans sa tâche d'interprétation propre et qui communique avec l'autre au moyen d'annotations. Le **troisième chapitre** constitue notre proposition de système pour l'interprétation collaborative de séries temporelles. Sa conception prend la forme d'un système multi-agents. Les séries temporelles étant des données complexes à traiter, nous détaillerons les modèles des données et les méthodes algorithmiques qui ont été utilisés. Le **quatrième chapitre** présente les évaluations du système. Le **cinquième et dernier chapitre** discute des propositions théoriques et techniques et présente les perspectives ouvertes par ce travail.

Chapitre 1

Amélioration du monitoring en unité de soins intensifs

L'objectif de ce chapitre est d'introduire la nécessité de proposer une nouvelle approche pour la construction de modèles d'évènements et de scénarios dans le contexte du monitoring de patients en unité de soins intensifs (USI).

Nous présentons les systèmes de monitoring à base de modèles comme des outils d'aide au diagnostique, au contrôle et à l'information de l'état du système.

Dans le contexte du monitoring de patients en USI, on constate le besoin d'amélioration des systèmes actuels ce qui implique l'utilisation de modèles complexes du système, en particulier des modèles d'évènements et de scénarios. Mais ces connaissances sont difficiles à extraire par des méthodes automatiques et difficiles à formaliser par les cliniciens.

Pour permettre l'amélioration du monitoring de patients en USI, l'objectif est donc de faciliter la construction de modèles d'évènements et de scénarios.

Le contexte du monitoring de patients en Unités de Soins Intensifs (USI) est à l'origine de cette thèse. La nécessité d'améliorer les systèmes actuels est aussi importante qu'elle est difficile. Il nous semble indispensable d'une part de positionner notre travail par rapport à cette question et d'autre part de comprendre la complexité pour justifier l'approche qui va être défendue. Nous présentons donc dans ce chapitre la problématique de l'amélioration des systèmes de monitoring de patients en USI.

A Monitoring de patients en Unités de Soins Intensifs (USI)

A.1 Principes du monitoring d'un système dynamique

Un système dynamique (SD) est un système¹ (mécanique, physiologique, chimique, ou encore un logiciel, une société ...) dont l'*état* évolue au cours du *temps*. Dans la pratique réelle, celui-ci ne peut être dissocié de la perception qu'il est possible d'avoir du système (en particulier dans le cas de systèmes très complexes tels que ceux auxquels nous aurons à faire). L'*état* et le *temps* sont perçus grâce à une instrumentation du SD qui les retranscrit plus ou moins fidèlement. Pour nous, l'état, ici sans italique, d'un SD à l'instant t sera un ensemble de valeurs de paramètres mesurés à ce même instant sur le SD par une instrumentation donnée.

Lorsqu'on observe l'évolution dans le temps du SD en fonctionnement, on échantillonne (à plus ou moins haute fréquence, régulièrement ou non) les mesures effectuées sur les différents paramètres. Chacun des paramètres conduit à la production d'une série temporelle. L'ensemble des séries temporelles, constituant une série temporelle multivariée, est la trace du fonctionnement du SD pendant une durée donnée.

En fonction de la nature des instruments et des paramètres observés, les valeurs acquises peuvent être numériques ou bien symboliques². Dans le cas des valeurs numériques, les valeurs prises par un paramètre sont réelles (*e.g.* température d'un four). La série d'observations effectuées dans le temps sera dénommée "série temporelle". Dans le cas des valeurs symboliques, les valeurs prises par un paramètre sont discrètes et on leur associe des symboles (*e.g.* "Système en marche"/ "Système à l'arrêt"). L'observation dans le temps de ce paramètre donnera lieu à une série temporelle *symbolique*.

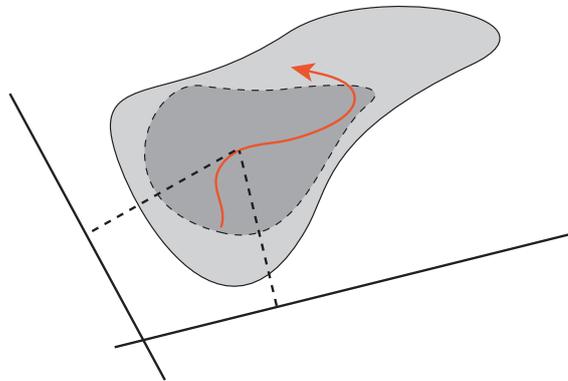


FIG. 1.1.

Représentation de l'évolution de l'état d'un SD (en orange) dans un espace d'états (en gris) à deux dimensions. Le gris sombre représente la zone des états acceptables pour le SD.

L'*état* du système à l'instant t peut être représenté par un point dans un espace abstrait des états possibles. L'état mesuré correspond à une projection de l'*état* sur des axes caractéristiques des instruments de mesure. Cette schématisation permet de représenter l'évolution d'un système dynamique dans un espace d'états possibles (en discrétisant un axe, la projection permet de représenter des paramètres discrets). La figure 1.1 illustre l'évolution de l'état d'un système en fonction du temps dans l'espace des états possibles à deux dimensions.

¹Je ne m'aventurerai pas à proposer une définition d'un *système*, le lecteur intéressé pourra trouver des réflexions sur ce point dans [136]

²Les systèmes qui ne se réduisent pas strictement à un fonctionnement continu ou à un fonctionnement par événements discrets sont dits hybrides [92].

Les systèmes dynamiques qui sont le centre d'intérêt du monitoring sont de plus en plus complexes : centrales nucléaires, véhicules, réseaux de télécommunications, patients, etc. On demande à ces systèmes d'être de plus en plus fiables afin de réduire les conséquences liées à leur dysfonctionnement. Le besoin d'activité de surveillance, de diagnostic (détection, localisation, identification de défaillances), de réparation et de reconfiguration se fait donc de plus en plus ressentir.

Le monitoring du système consiste à surveiller l'état d'un système et son évolution. D'après Basseville et al. [19], les trois raisons qui amènent à la surveillance de systèmes dynamiques sont :

- la *conduite* qui permet d'assurer que l'état du SD reste dans une zone de l'espace des états possibles qui soit optimale, souhaitable ou requise ;
- la proposition d'un *service* permettant de détecter et diagnostiquer les événements indésirables ;
- la *maintenance* corrective lorsqu'un dysfonctionnement a été identifié, ou préventive lorsqu'une action sur le système peut éviter d'engendrer des dysfonctionnements.

A.1.1 Des systèmes automatiques de monitoring pour aider les opérateurs

L'objectif des *systèmes automatiques de monitoring* est d'aider un opérateur humain dans la surveillance d'un SD et le suppléer dans les prises de décisions qui permettront un fonctionnement optimal du système à court et long termes.

Devant le nombre de systèmes à surveiller et la masse d'informations de plus en plus importante dont il faut tenir compte pour chacun d'eux, le monitoring est devenu une tâche monumentale que les opérateurs ne peuvent assurer seuls. Il est donc nécessaire d'automatiser une partie du monitoring.

Pour mettre en évidence cet accroissement de l'automatisation de la surveillance, nous pouvons prendre l'exemple de la détection d'un problème dans la mécanique d'une voiture. Il y a quelques années la surveillance de l'état d'une voiture (par exemple une 2CV) se faisait majoritairement par son propriétaire au son du moteur et à la vérification de quelques indicateurs (*e.g.* le niveau d'huile). L'automatisation du monitoring était donc nul. Jusqu'à il y a quelques années, les propriétaires d'une voiture tombant en panne pouvait apercevoir un voyant rouge et le garagiste expérimenté diagnostiquait la panne. L'utilisation des capteurs pour faire apparaître à l'utilisateur une image du fonctionnement du système et le lancement d'alarmes simples sont un premier pas vers le monitoring. Aujourd'hui, une voiture récente indique à son conducteur la nécessité de faire une vidange (parfois même l'anticipe), et le diagnostic de l'état de la voiture est réalisé par le garagiste grâce des systèmes électroniques complexes.

L'utilisation de systèmes automatiques permet de réduire les coûts économiques de la surveillance de système, et ce :

- en permettant de monitorer un plus grand nombre de paramètres (sur un même système ou pour plusieurs systèmes) avec les mêmes moyens humains,
- en réduisant le personnel nécessaire pour réaliser une tâche de surveillance,
- en réduisant les temps d'immobilisation d'un système endommagé par de meilleurs diagnostics,
- en réduisant le nombre de pannes en les prévenant.

Mais l'automatisation a également des apports bénéfiques sur la qualité du monitoring qui est réalisé puisque cela permet :

- d'assurer une aide à la prise de décision intégrant un grand nombre de variables pour
 - éviter les erreurs humaines : en recoupant les informations du système avec les actions entreprises par l'opérateur, on peut anticiper les dangers potentiels pour le système ;
 - améliorer la précision des diagnostics : en s'aidant d'une instrumentation de plus en plus précise, les diagnostics qui peuvent être réalisés le sont d'autant plus ;
 - optimiser les décisions (plus complètes, plus rapides) : l'information qui peut être traitée est globalement plus riche que ce que pourrait traiter un opérateur seul. En intégrant plus de facteurs dans la prise de décision celle-ci est plus complète. De plus, le traitement de ces informations est réalisé très vite par des machines de plus en plus performantes accélérant la prise de décision en proposant rapidement les bonnes actions à mener. Ceci permet de réduire les temps de réaction face à une situation critique.
- d'assurer une certaine constance de la surveillance : la tâche de surveillance est fastidieuse pour un humain et celui-ci ne peut conserver le même niveau d'attention sur de longues durées.
- d'améliorer la connaissance à long terme du système : les enregistrements réalisés des paramètres

permettent de rétrospectivement analyser le fonctionnement (ou le dysfonctionnement) du système pour en améliorer la connaissance.

Bien que beaucoup de facteurs plaident pour l'automatisation de la tâche de monitoring, la prise de décision reste très souvent l'apanage des opérateurs humains. Le système ne permet de faire que des constatations (charge à l'opérateur de décider d'une intervention), des recommandations pour une action ou bien d'initier des actions en prenant soin d'en informer les opérateurs et de s'effacer devant les décisions de ceux-ci. Dans des cas exceptionnels qui correspondent à des situations d'urgences dans lesquelles les opérateurs humains sont estimés comme hors d'état d'action, le système peut prendre le contrôle du système au-dessus des recommandations des opérateurs (*e.g.* système anti-crash pour les avions de combat, système de sortie des barres de combustibles dans les centrales nucléaires).

A.1.2 Tâches réalisées par un système de monitoring

Les trois grandes fonctions du monitoring présentées dans [19] sont : "Voir, Comprendre et Agir". L'acquisition des données par une instrumentation appropriée permet de percevoir l'état du système. À cette étape on ajoute des méthodes de pré-traitements des données (renormalisation, rééchantillonnage, suppression de bruits et de signaux parasites, ...). Les séries temporelles qui sont extraites de ces deux étapes doivent ensuite être interprétées par le système. Pour cela, les informations les plus pertinentes sont sélectionnées des données prétraitées et un diagnostic de l'état du système est réalisé par un processus de décision.

La chaîne de traitement représentée sur la figure 1.2 comprend donc quatre grandes étapes : l'acquisition des paramètres, le pré-traitement des données, la sélection des informations et la prise de décision.

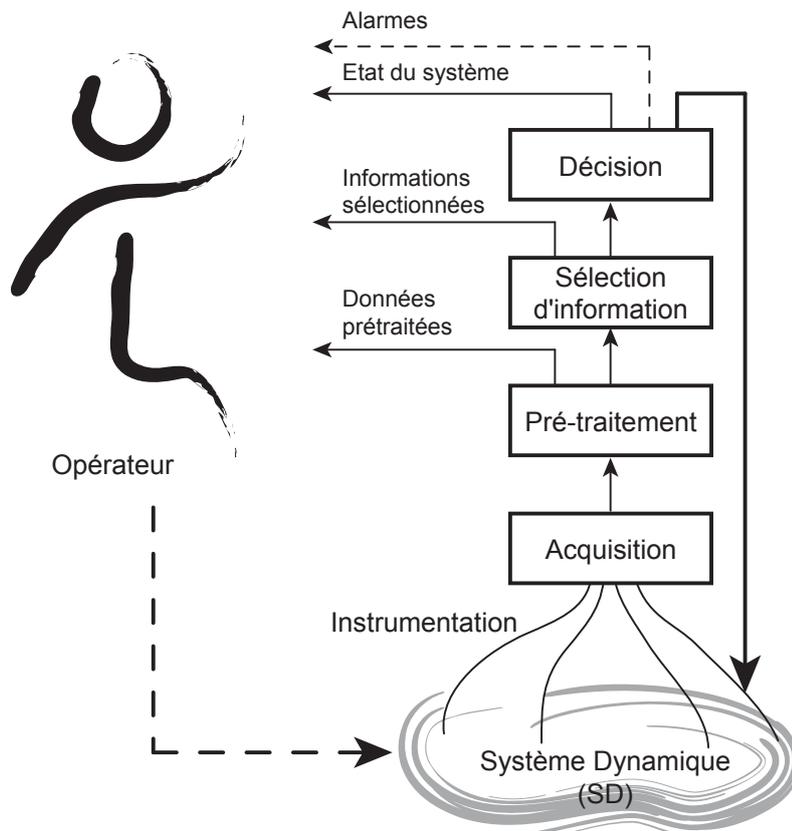


FIG. 1.2. Chaîne de traitement d'un système de monitoring : De l'acquisition des signaux jusqu'à la décision.

À chacune de ces étapes, l'opérateur reçoit des informations. Les données prétraitées représentent une grande masse d'informations qui n'est généralement pas utilisée. Elles nécessitent d'être épurées

spécifiquement pour un opérateur. À ce niveau, une sélection de l'information la plus utile à l'opérateur doit être retranscrite de manière compréhensible pour qu'il suive l'évolution du SD. Enfin, le processus de décision permet au système de proposer un diagnostic de l'état du système et d'alerter l'opérateur d'une situation critique en vue de l'inciter à agir pour modifier l'état du système.

À partir de la figure 1.2, il nous apparaît intéressant d'identifier trois grandes tâches à réaliser par le système automatique dans l'objectif d'aider un opérateur dans la surveillance du SD :

- le diagnostic automatique de l'état du SD par la perception et l'interprétation des paramètres suivis (fonctionnalités “voir” et “comprendre”) (Flèches montantes),
- l'information à l'opérateur qui pourra agir sur le système (Flèches horizontales),
- l'action du système de monitoring qu'il peut réaliser lui-même sur le SD : le système “Agit” directement sur le SD afin qu'il s'adapte à de nouvelles conditions de fonctionnement (Flèche descendante pleine).

Diagnostic de l'état du SD à base de modèles Le diagnostic de l'état du SD consiste à réaliser une décision à l'instant t sur l'état du système [42, 137, 205]. Les performances du système de monitoring sont liées d'une part à sa capacité à ne pas faire d'erreurs sur l'état du système attribué par le processus de décision, c'est la spécificité et d'autre part être capable de détecter un maximum des occurrences des états recherchés du système, c'est la sensibilité³. Pour réaliser le diagnostic, le système de monitoring s'appuie (1) sur les informations qui peuvent être recueillies sur le SD et (2) sur des modèles⁴ du système. On parle alors de diagnostic basé sur les modèles [72].

Les informations recueillies à partir du système sont bien évidemment les données brutes à l'instant t et éventuellement aux instants précédents, mais également les résultats du pré-traitement et de la sélection des données prétraitées.

L'utilisation des seules valeurs à l'instant t pour estimer l'état du système à t constitue le monitoring le plus simple dont les performances attendues sont assez réduites. Pour assurer une meilleure finesse et réduire également le taux d'erreur, il est généralement intéressant de ne pas se concentrer uniquement sur l'état du SD à t , mais d'intégrer également la tendance de son évolution. Dans ce but, les valeurs brutes des paramètres qui sont à considérer intègrent les valeurs aux instants précédents qui sont facilement disponibles.

Dans le domaine médical, le pré-traitement usuel des séries temporelles est la construction d'abstractions temporelles des données. Ce type de pré-traitement construit des informations de plus haut niveau d'abstraction qui intègre la temporalité du système, *e.g.* sous la forme de tendances. Pour une revue complète des méthodes d'abstraction temporelle, le lecteur intéressé pourra consulter les premiers chapitre de la thèse de A.S. Silvent [194] ou bien l'article récent de Stacey et McGregor [198].

La complexité des pré-traitements et des sélections peut ou non être contrainte par le besoin de réaliser le diagnostic dans un temps réduit (cas des décisions en temps réel). Une solution à ce problème est l'utilisation de “capteurs intelligents”. Les pré-traitements et les sélections ne sont plus centralisés mais distribués dans les capteurs eux-mêmes qui fournissent alors uniquement les données pré-traitées et filtrées.

Les modèles sont des connaissances expertes sur le système qui ont été formalisées et intégrées au système de monitoring pour que celui-ci identifie l'état du système à partir des informations disponibles. Un grande quantité de formalismes peut servir à cette modélisation : les réseaux de neurones [138], les méthodes à noyaux (*e.g.* SVM [36]), les modèles probabilistes (*e.g.* réseaux Bayésiens [36, 138], modèles possibilistes [174], ...), les formalismes logiques (*e.g.* [49]), les modèles causaux [67]) ...

Deux types de modèles peuvent être construits :

- les modèles de faute qui associent un ensemble d'états spécifiés et connus comme étant non-désirés avec un comportement du système de monitoring (une information à l'opérateur et/ou une action sur le SD) ;
- les modèles de fonctionnement qui permettent de définir les états acceptables du système en dehors

³Nous illustrerons par la suite les notions de spécificité et sensibilité sur le déclenchement d'alarmes.

⁴De même qu'il est difficile de parler de *système*, la notion de *modèle* est difficile. Je m'arrête ici sur une définition proposée par Dupuy [89] : “Le modèle est une forme abstraite qui vient à s'incarner et se réaliser dans les phénomènes”. À la vue de l'importance de cette notion, on reviendra par la suite sur celle-ci.

desquels il y a un problème.

Dans le premier cas, les décisions qui pourront être fournies seront riches (on pourra informer l'opérateur du cas critique identifié), mais cela nécessite d'être capable de définir les zones de l'espace d'état qui correspondent à des fautes. Si ces zones sont mal définies, on peut manquer d'informer l'opérateur de nombreuses situations critiques non envisagées initialement. Le risque d'erreur est donc important. On a alors une sensibilité réduite, mais une bonne spécificité. Dans le second cas, les alarmes ne seront que peu informatives (Le diagnostic n'est que très partiel et impose à l'opérateur l'identification de la nature de l'alarme), mais en contrepartie on peut garantir que tant qu'il n'y a pas d'alarme le système se comporte bien. On a alors une trop forte spécificité, mais une sensibilité accrue.

Bien évidemment, la solution consiste souvent à allier les deux pour avoir un compromis intéressant entre la capacité de fournir des alarmes informatives avec peu d'erreurs.

Restitution de l'information Le processus de monitoring est un processus d'aide à la décision pour des utilisateurs chargés d'assurer le bon fonctionnement et le maintien du SD dans un tel état. En conséquence, l'information recueillie pendant le processus de décision (les données et les diagnostics réalisés) est fournie à l'utilisateur.

Deux façons communes sont proposées pour restituer l'information sur l'état du système. De manière continue, l'information peut être mise à disposition immédiate de l'observateur qui réalise à son tour une surveillance du SD avec ses capacités cognitives propres. De manière ponctuelle, il s'agit de transmettre des messages spécifiques (alarmes, informations, ...) uniquement lorsque c'est nécessaire.

L'installation d'un système de monitoring permet de centraliser les informations pour les mettre à disposition immédiate d'un observateur. Ces données peuvent être ainsi visualisées en temps réel afin de restituer l'état du système à l'observateur. Ce type de tâche est utile lorsque l'opérateur ne voit pas directement le SD observé : le SD est trop petit (*e.g.* fonctionnement d'une puce électronique), trop distant (*e.g.* rover marsien), dans des zones inaccessibles aux humains (*e.g.* cœur d'une centrale nucléaire), ... La masse de données qui peuvent être engendrées par l'instrumentation d'un système peut être très grande et les données brutes difficilement utilisables par un opérateur humain. Donc pour certains systèmes de monitoring la visualisation passe par une étape de sélection des informations la plus pertinente et/ou la transformation des données en des informations de plus haut niveau d'abstraction (par des méthodes similaires à celles qui permettent de faire le diagnostic du système) afin de présenter à l'observateur une information plus facilement exploitable. Par exemple sur la figure 1.3, le système de monitoring cardiaque donne directement le rythme cardiaque d'un patient et pas uniquement le signal brut sur lequel cette information capitale pour un médecin n'apparaît pas immédiatement.



FIG. 1.3.

Moniteur Cardiaque. Le moniteur affiche les signaux qui sont recueillis sur le patient mais également des informations résumées extraites de ces signaux (comme la fréquence cardiaque) afin de faciliter le diagnostic du clinicien.

Les alarmes, qu'elles soient sonores ou visuelles, sont lancées lorsque le processus de décision conduit à

identifier grâce à des modèles d’alarmes une situation critique nécessitant l’information ou l’intervention plus ou moins rapide de l’opérateur. L’alarme a un rôle de focalisation de l’attention de l’opérateur. Par conséquent, pour atteindre son objectif elle doit pointer aussi précisément que possible la cause probable du dysfonctionnement et le niveau de gravité. De nouveau, plus le système produira des informations riches et précises à l’opérateur, moins le travail d’interprétation lui sera difficile et plus il sera efficace.

La qualité d’un système de monitoring est généralement évaluée selon sa spécificité et sa sensibilité à lancer des alarmes. Pour les calculer, on évalue le nombre des quatre situations possible d’alarmes :

- les vrais positifs (VP) : Une alarme a été lancée sur un cas le nécessitant ;
- les vrais négatifs (VN) : Une alarme n’a pas été lancée dans une situation ne le nécessitant pas ;
- les faux positifs (FP) : Une alarme a été lancée sur une situation ne le nécessitant pas ;
- les faux négatifs (FN) : Une alarme n’a pas été lancée sur un cas le nécessitant ;

Dans les deux premiers cas, le diagnostic du système a été correct. Le cas des faux négatifs est la situation où le système de monitoring n’a pas réussi à diagnostiquer une situation critique et peut conduire à mettre le système surveillé en dysfonctionnement. Le cas des faux positifs est une situation où une alarme intempestive a été lancée par le système de monitoring. La spécificité (S_p) et la sensibilité (S_e) du système de monitoring sont alors des valeurs entre 0 et 1 calculées par les formules suivantes :

$$S_p = \frac{VN}{VN + FP} = 1 - \frac{VP}{VN + FP}, \quad S_e = \frac{VP}{VP + FN}$$

La sensibilité est le nombre d’alarmes correctement lancées parmi toutes les situations critiques et la spécificité est donc le “nombre” d’alarmes qui n’ont pas été déclenchées à juste titre parmi tous les fausses alarmes. Une spécificité forte indique donc que beaucoup d’alarmes lancées ne sont pas significatives et conduit l’opérateur à ne plus réagir aussi rapidement à celles-ci. Une sensibilité faible indique que de nombreuses situations critiques ne sont pas détectées par le système de monitoring et conduit l’opérateur à ne plus avoir confiance dans cet outil. Il est usuel de s’intéresser aux quantités S_e , $1 - S_p$, et non S_p , qui indiquent les bonnes performances du système lorsqu’elles sont toutes les deux proches de 1 ⁵⁶.

Contrôle du fonctionnement du système dynamique Le fonctionnement définit dans la figure 1.2 met en évidence deux boucles de contrôle du SD dans lesquels le système de monitoring intervient comme aide à l’opérateur soit directement sur SD. Dans le premier cas, une boucle se ferme par les actions entreprises par l’opérateur à la suite d’une information réalisée par le système de monitoring. Dans le second cas, une boucle se ferme par des actions réalisées par le système lui-même. Le système est alors également effecteur (C’est le cas par exemple des ventilateurs artificiels, ou bien des systèmes de filtrage de sang pour réaliser des dialyses). Cette seconde boucle (boucle de rétro-action ou de *feedback*) permet d’assurer une régulation *automatique* du fonctionnement du SD (Fonctionnement en boucle fermée).

De nouveaux des modèles de contrôle (on les oppose alors aux modèles d’alarme) peuvent être utilisés pour définir la réaction appropriée que la machine doit avoir en réponse à un état constaté. Dean et Wellchman [73] distinguent trois niveaux de contrôle :

- Au niveau L1, des boucles de contrôle locales et réactives pour permettre l’asservissement d’une variable ;
- Au niveau L2, l’utilisation de modèles plus complexe du fonctionnement de système vise à un contrôle global ;
- Au niveau L3, l’évolution du système est planifiée pour permettre de le conduire dans l’état désiré.

En conclusion, les systèmes de monitoring peuvent être vus comme une chaîne de traitements qui

⁵De plus, on peut évaluer la précision globale du système par la proportion de prédictions correctes réalisées par le système, A (Accuracy) :

$$A = \frac{VP + VN}{VP + VN + FP + FN}.$$

⁶Dans la pratique, il peut être difficile de quantifier VN , le “nombre” de situation ne nécessitant pas de lancer des alarmes. Le couple de mesures Rappel/Précision utilisé dans le domaine de la Recherche d’Information peut alors être employé :

$$P = \frac{VP}{VP + FP}, \quad R = S_e = \frac{VP}{VP + FN}$$

Le rappel correspond à la sensibilité et la précision indique alors le nombre d’alarmes correctement lancées parmi toutes les alarmes lancées. Il n’y a pas de relation linéaire entre la précision et la spécificité.

permet de réaliser des décisions à base de modèles et à partir des données des capteurs. Les tâches qui incombent à un tel système sont la réalisation du diagnostic, l’information à l’opérateur et la régulation du fonctionnement du système.

A.2 Le monitoring de patients en USI

Les Unités de Soins Intensifs (USI) sont un contexte d’applications privilégié et de plus en plus exigeant pour des systèmes de monitoring. Les outils d’aide à la décision sont déjà implantés dans les unités de soins et les cliniciens en tiennent compte dans leur pratique [85]. Les différentes situations qui peuvent être rencontrées requièrent une technologie et des traitements de pointe pour les systèmes de monitoring de plus en plus complexes sous une surveillance continue, et le personnel médical spécialisé a besoin d’être suppléé dans cette surveillance pour se concentrer sur ses tâches les plus importantes, *i.e.* les actes médicaux.

Imhoff et *al.* [121] reprennent la terminologie de l’analyse de processus industriels pour définir les moniteurs : “Monitors are sensors that provide health care professionals with patient-related data and information that support those professionals in making appropriate therapeutic decisions”. Hudson (cité dans Pierson [170]) définit le monitoring en soins intensifs ainsi “Monitoring is making repeated or continuous observations or measurements of the patient, his or her physiological function and the function of life support equipment, for the purpose of guiding management decisions, including when to make interventions and assessment of those interventions”. Il recoupe ainsi les grandes tâches présentées dans la section précédente.

A.2.1 Unités de soins intensifs

Nous regroupons sous le terme d’unité de soins intensifs (USI) les unités d’anesthésie et de réanimation (médicales ou chirurgicales), les unités de surveillance continue et les unités de soins intensifs proprement dites. Ces dernières peuvent être spécialisées pour un type de patient (*e.g.* USI Cardiologique) ou non. Chacune de ces unités est régie par des contraintes et recommandations spécifiques. Mais dans les trois cas, il s’agit d’unités ayant peu de lits en charge, situées dans des locaux spécifiquement aménagés pour accueillir un équipement matériel spécialisé et proposant un encadrement 24/24h par du personnel médical spécifiquement formé et expérimenté. Ils contribuent ainsi à la prise en charge des cas de patients généralement les plus graves et qui ont besoin d’une surveillance constante que ce soit pour les diagnostiquer ou pour permettre de réagir au plus vite à des détresses vitales.

Au sein de l’organisation même de ces unités, les systèmes automatiques de monitoring ou d’assistances ont une position centrale. Ainsi, les signaux vitaux sont redirigés vers une salle qui centralise les informations de l’ensemble des patients de sorte que le personnel soignant puisse être rapidement informé de l’état de l’ensemble des patients depuis un même lieu.

Les systèmes de monitoring qui sont utilisés sont principalement conçus pour émettre des alarmes sonores lorsque l’état du patient n’est pas diagnostiqué comme bon. Ces alarmes peuvent être centralisées également, mais elle nécessiteront une intervention au pied du lit du patient pour être désenclenchées.

Un patient arrivant en USI est généralement dans un état instable. Son évolution peut être brutale, et dans l’état actuel des choses, difficilement prévisible. Ceci est moins vrai pour les patients qui sont en réanimation chirurgicale. Les principales pathologies rencontrées dans les unités de soins intensifs sont :

- Insuffisance respiratoire aiguë nécessitant une assistance respiratoire;
- État de choc lié à une insuffisance cardiaque nécessitant une pompe ou un coeur artificiel pour soutenir la circulation, l’administration de médicaments puissants et un monitoring complexe;
- Transplantations;
- Comas graves ou crises épileptiques prolongées;
- Traumatismes et brûlures extrêmes;
- Défaillances d’organes multiples.
- etc.

Si on s’intéresse à un patient comme système dynamique pour re-situer la tâche de son monitoring dans le contexte général présenté précédemment, il peut être considéré comme *complexe* par le caractère

non-linéaire et non prévisible de son évolution. Il faut indiquer qu'il s'agit là d'un système intrinsèquement beaucoup plus difficile à surveiller que n'importe quel système technique. En particulier, il faut pouvoir proposer des méthodes qui tiennent compte de la disparité qui existe entre chaque patient. De plus, contrairement à des systèmes techniques pour lesquels les fonctionnements sont généralement bien spécifiés à la construction, il est difficile d'extraire des connaissances physiologiques des mécanismes généraux applicables aux cas pathologiques diversifiés rencontrés dans les USI.

A.2.2 Équipement d'une salle d'USI

Les paramètres physiologiques Le monitoring du patient commence par l'acquisition de paramètres qui vont permettre de diagnostiquer son état physiologique. Cependant, la complexité du "système patient" est telle qu'il ne faut pas espérer pouvoir identifier l'état physiologique de celui-ci avec ces quelques paramètres. Ainsi, les paramètres qui sont principalement considérés sont ceux qui peuvent être mis en relation assez directe avec les fonctionnalités physiologiques majeures d'un patient (respiratoires, cardiaques, rénales, ...). Ces fonctionnalités étant elles-mêmes intimement liées au pronostic vital du patient.

L'acquisition de ces paramètres est soumise à une exigence de leur représentativité de la fonction physiologique à laquelle ils sont associés. Cela nécessite l'emploi d'instrumentations et de méthodes pointues permettant de les extraire au milieu des autres signaux émanant du corps et de prendre en compte les différents bruits auxquels ils sont soumis.

Les deux fonctionnalités centrales sont la respiration et l'activité cardiaque de sorte à pouvoir assurer un approvisionnement en oxygène dans l'ensemble du corps et l'extraction du dioxyde de carbone. Les données importantes de la respiration sont le débit, la pression et la fréquence respiratoire (paramètres liés entre eux), et celles de l'activité cardiaque sont les pressions et les débits de sang dans le coeur.

En pratique médicale, un médecin s'appuie couramment sur des informations beaucoup plus riches que les seuls paramètres physiologiques pour dresser l'état général d'un patient. Par exemple, la posture générale d'un patient, son teint, son activité physique, les remarques du patient sur son propre état lors de discussions... Et des paramètres contextuels peuvent également influencer sur l'état du patient (*e.g.* la température de la chambre, la présence d'une personne, ...) et pourraient être également considérés. Dans une USI, les patients sont dans des états limites, hors des schémas bien connus, et ils sont généralement équipés avec de nombreux moniteurs. Si une certaine intuition basée sur l'expérience des cliniciens reste pertinente, les cliniciens s'appuient de plus en plus sur les signaux et les moniteurs pour juger de l'état du patient.

Équipement d'une salle d'USI Chaque patient (individuellement dans une salle de soins) est équipé selon [121] de moniteurs et équipements d'assistance standards (associés aux paramètres vitaux) et de certains équipements spécifiques à son état et sa thérapie :

- Ventilateur artificiel pour l'assistance respiratoire. Mesure et surveille les débits, fréquences et volumes respiratoires ;
- Oxymètre. Mesure de la SpO_2 , pourcentage d'oxygène (O_2) dans le sang.
- Capnographe. Mesure le pourcentage de dioxyde de carbone (CO_2) expiré ;
- Moniteur cardio-vasculaire ou Électrocardiogramme (ECG). Mesure et surveille les pressions et débit sanguins cardiaques ainsi que les fréquences cardiaques,
- Autres appareils de mesure physiologiques non invasifs : Température, pression au brassard, ...
- Pompes et perfusions pour l'administration de médicaments
- Appareils d'échange de fluides physiologiques
- ...

Certains de ces appareils sont des effecteurs, c'est-à-dire qu'ils ont des actions effectives sur le patient (*e.g.* un ventilateur), et d'autres ne sont que des moniteurs, et ne font qu'observer des paramètres. Dans les deux cas, ils sont associés avec des mécanismes automatiques de monitoring qui permettent de suivre un paramètre et de déclencher des alarmes au besoin.

Le contexte des unités de soins intensifs est particulièrement difficile pour la problématique de monitoring. L'utilisation de systèmes automatiques de surveillance s'est principalement focalisé sur la possibilité d'émettre des alarmes (sonores) lorsque la situation est jugée critique par le système. Chacun des équipements peut émettre ses propres alarmes (la prise de décision n'est pas centralisée). M.C. Chambrin [51]

dénombrer ainsi près de 40 sources possibles d’alarmes depuis un même patient.

Il est reconnu que l’efficacité de ces alarmes est réduite [26, 54, 133, 162, 213] car leur taux de fausses alarmes est élevé⁷. Souvent, les algorithmes utilisés dans les moniteurs sont simples [121, 210]. Ils se contentent généralement de déclencher des alarmes lorsque la valeur d’un paramètre dépasse un seuil fixé *a priori*. Les algorithmes sont ainsi très sensibles aux bruits dans les données captées (*e.g.* les capteurs d’oxymétrie (SpO_2) placé au bout d’un doigt d’un patient peuvent facilement se débrancher⁸).

La configuration des systèmes de monitoring par l’équipe médicale⁹, par exemple les seuils au delà desquels un paramètre physiologique ne doit pas aller, est également critique pour le déclenchement des alarmes et donc pour l’efficacité du système de monitoring. Si cette configuration est mal faite, elle peut même devenir néfaste au patient [71].

De plus, la conception même de ces systèmes cherche à avoir une sensibilité maximale et ceci au détriment des prises de décisions précises [96]. Dans le principe où le clinicien est au final seul juge de la criticité de l’état d’un patient, les systèmes de monitoring et leur configuration tendent à déclencher facilement les alarmes car en premier lieu, il faut éviter d’en rater. Le nombre d’alarmes faussement positives est en conséquence très élevé.

La réduction du taux de fausses alarmes est un enjeu majeur pour l’amélioration des soins et des conditions de travail dans les unités de soins intensifs.

B Amélioration des systèmes de monitoring de patients en USI

Nous dessinons maintenant un panorama des améliorations qui peuvent être apportées aux systèmes de monitoring pour les USI. À partir de ce constat de la nécessité d’amélioration des systèmes de monitoring, on peut envisager de s’attaquer principalement au problème des fausses alarmes. Mais nous présentons également les améliorations qui sont en cours de développement pour enrichir ces systèmes par de nouvelles fonctionnalités qui aideront les cliniciens dans leur pratique quotidienne. Ceux deux types d’amélioration permettent d’envisager les systèmes de monitoring de demain qui seront plus intelligents et qui réaliseront des tâches plus complexes que le “simple” envoi d’alarmes. Nous présentons ensuite, les voies techniques envisagées pour améliorer tout ou partie de ces performances par des méthodes de traitement de l’information¹⁰.

B.1 Amélioration des performances du monitoring

Les besoins d’amélioration des performances des systèmes de monitoring actuels mis en évidence par les études cliniques sont liés tout d’abord à l’amélioration de la spécificité et la sensibilité de chaque étape de la chaîne de traitement jusqu’à la décision, mais également à la nécessité d’obtenir des alarmes plus significatives, *i.e.* fournir une information efficace aux cliniciens, et de déclencher celles-ci au plus tôt.

B.1.1 Amélioration de la chaîne de traitement de l’information

La réduction du taux de fausses alarmes correspond à un problème d’amélioration de la précision du diagnostic, mais elle dépend fortement de la capacité de la chaîne de traitement de l’information qui la précède. Cette problématique touche toutes les étapes par lesquelles les données sont acquises et transformées :

- Instrumentation : L’information recueillie à la base est essentielle. Elle doit être riche (*i.e.* diversifiée et informative) pour pouvoir représenter au mieux l’état réel du patient. La diversité des sources doit permettre de multiplier les points de vue afin qu’ils puissent être croisés et d’en tirer des informations de meilleure qualité.

⁷Le taux de fausses alarmes dans une unité de soins intensifs est de l’ordre de 80% [51, 130, 213], et seulement 5% des alarmes entraînent une modification de la prise en charge d’un patient [54].

⁸Un capteur d’oxymétrie du sang par spectrophotométrie identifie la quantité d’oxygène dans le sang à partir de sa couleur dans des capillaires de l’extrémité du doigt ou du lobe d’oreille.

⁹En France, des équipements tels que les ventilateurs ne peuvent être manipulés que par les cliniciens eux-mêmes.

¹⁰Nous nous intéressons dans cette thèse au traitement de l’information, mais d’importants travaux pour l’amélioration des systèmes de monitoring sont également menés sur l’instrumentation elle-même.

- Méthodes de pré-traitement : Elles doivent permettre de tirer au mieux partie de la masse de données en focalisant les traitements avals sur l’information pertinente sans pour autant en élaguer trop pour éviter de faire baisser la sensibilité du système.
- Modèles sur lesquels sont réalisés les diagnostics : Les prises de décision doivent tenir compte de plus d’informations et ainsi rendre les diagnostics plus précis. Il faut pour cela s’appuyer sur des modèles de fautes plus complets que ceux actuellement utilisés.
- Algorithmes de décision en temps réel : Une des limites reste toujours la puissance de calcul des moniteurs qui sont utilisés. Les algorithmes et leurs supports matériels doivent être étudiés pour faire fonctionner des méthodes de diagnostic en temps réel.

B.1.2 Alarmes plus informatives

Le monitoring étant une tâche principalement d’aide à la surveillance et à la décision pour le personnel médical, les moniteurs doivent s’attacher à fournir une information plus riche. Actuellement, les alarmes sont classées en trois catégories [51] : les alarmes de haute priorité indiquant une situation d’urgence nécessitant une intervention immédiate, les alarmes de moyenne priorité indiquant une situation dangereuse nécessitant une intervention rapide et les alarmes de basse priorité attirant l’attention du personnel clinique sur la nécessité d’une intervention (*e.g.* une défaillance sur un capteur). En fonction de leurs catégories, les alarmes varient par leur “mélodie” et leur fréquence. Cependant, les appareils de monitoring qui distinguent les types d’alarmes sont rares. S’ils sont rares, c’est que la production d’alarmes ou d’informations plus significatives nécessite d’être capable de distinguer plus de scènes dans le processus de décision que les trois niveaux de criticité.

En pratique, on peut estimer que les alarmes qui sont émises sont peu informatives pour le personnel médical [91]. L’aide à la compréhension de l’information qui est fournie par le système de monitoring ou une indication de la réaction attendue à la suite d’une alarme dans le cadre d’un protocole permettrait d’améliorer l’utilisation de ceux-ci [71]. Ceci passe également par l’amélioration des interfaces graphiques qui permettent d’enrichir les seules informations des signaux biomédicaux. Dans ce but, Wachter et *al.* [220] proposent un système pour représenter graphiquement les variables respiratoires. Le système a été estimé utile par le personnel soignant en complément des moniteurs des signaux biomédicaux.

La prise en compte de l’incertitude du système lorsqu’il lance une alarme peut également être une information riche de sens pour que les cliniciens organisent au mieux leurs interventions, mais également pour qu’ils aient une confiance accrue dans le système. Les cliniciens accepteront plus facilement les erreurs du système si elles sont accompagnées d’une incertitude et seront plus à même de répondre rapidement à une alarme qualifiée de sûre par le système (à condition que ce type d’alarmes ait une forte spécificité).

La volonté d’envoyer des alarmes plus informatives ne doit pas devenir un inconvénient dans la pratique. Du fait de la grande variété des patients, obtenir des alarmes capables de distinguer plus de scènes critiques peut conduire à devoir paramétrer finement les futurs systèmes de monitoring. Le gain d’information doit justifier le temps de la configuration de l’outil, sinon on peut d’hors et déjà s’attendre à ce qu’il ne soit pas utilisé pleinement.

B.1.3 Déclencher au plus tôt les alarmes

Lorsqu’une alarme se déclenche, la situation critique s’est déjà passée et les interventions doivent être rapides. Un des objectifs d’amélioration des systèmes de monitoring est d’éviter d’aller jusqu’à cette situation critique et de pouvoir l’anticiper au plus tôt. Pour cela, le système doit être capable d’identifier des signes avant-coureurs d’un état critique. Ceci permettrait 1) d’éviter d’arriver à des scènes où le patient est en état critique, 2) de permettre au personnel de mieux gérer ses interventions lorsque plusieurs patients nécessitent leur attention.

B.2 Amélioration des services d’un système de monitoring

Au delà des fonctions de monitoring, les systèmes automatiques peuvent aider à améliorer les soins en proposant des services supplémentaires. Les services qui ont le plus d’utilité pratique sont ceux qui vont réduire le nombre d’interventions sur le système de monitoring (effecteur ou moniteur) pour en modifier le réglage. Par conséquent, un service utile pour le personnel clinique est en premier lieu une aide à la

configuration (*i.e.* paramétrage des moniteurs). Ensuite, l'automatisation de l'évolution du paramétrage des machines pour que celles-ci suivent un protocole défini sans qu'il faille nécessairement manipuler les appareils constamment.

B.2.1 Aide à la configuration ou à la re-configuration

Certains appareils tels que les ventilateurs sont des systèmes qui nécessitent une bonne expérience pour pouvoir être configurés efficacement et surtout qu'ils n'aient pas de conséquence physiologique grave sur le patient et que les alarmes soient pertinentes.

Des outils d'aide à la configuration doivent permettre d'accélérer le réglage d'un appareil, car si les systèmes sont amenés à se complexifier, il est certain que leur réglage sera également de plus en plus long. Si on veut que ces systèmes soient utilisés en clinique, il est donc nécessaire de rendre efficace ces réglages. La configuration pouvant être critique, l'introduction de systèmes de vérification des réglages peut conduire à éviter des erreurs en informant le personnel médical d'une configuration erronée.

À ma connaissance, il n'existe pas de projet qui vise à réaliser ce type d'améliorations des systèmes de monitoring. Le travail qui s'en rapproche le plus est la thèse de F. Portet [171] qui vise à la sélection automatique des signaux à considérer pour un système de monitoring cardiaque.

B.2.2 Suivi automatique de protocole

La mise en place de protocoles (*Guidelines*) est pointée par A.S. Silvent [194] comme une voie d'amélioration de la surveillance dans les USI. Ils permettent de mieux cadrer le contexte des données qui sont traitées par les systèmes automatiques. L'utilisation de cette connaissance facilite les traitements automatiques.

Pour les cliniciens, les protocoles leur donnent l'opportunité d'automatiser une succession de tâches. L'objectif étant de ne pas avoir à intervenir plusieurs fois sur un patient pendant le protocole, comme c'est le cas actuellement, mais que les systèmes effecteurs suivent d'eux-mêmes le protocole.

Par exemple, pour un patient ventilé, l'assistance mécanique ne peut être arrêtée brutalement. Dans ce cas, il faut passer par une étape de sevrage dans laquelle l'assistance ventilatoire doit être réduite progressivement pour que les muscles respiratoires reprennent d'eux-mêmes leur fonction. On sait que la durée du sevrage peut être raccourcie par l'utilisation de protocoles [142].

Le SmartCare [152] est un système à base de connaissances (un système expert utilisant des règles) qui automatise l'exécution d'un protocole de sevrage à un ventilateur artificiel. Le système SmartCare implémente la réduction *automatique* de cette assistance tout en assurant la fonction respiratoire. Un tel système permet ainsi d'éviter de fixer un clinicien au pied d'un patient ou de lui imposer de réduire brutalement l'aide respiratoire. Le système permet ainsi de réduire les temps de sevrage et d'hospitalisation d'un patient en USI.

B.3 Voies d'amélioration des systèmes de monitoring

L'amélioration des systèmes de monitoring implique l'amélioration des techniques de traitement des données une fois celles-ci acquises. Ces nouvelles techniques de traitement de l'information participent à la construction de systèmes de monitoring intelligents [155, 156, 161].

La difficulté des informations qui sont traitées réside dans leur temporalité. Comme le suggère l'article de Shahar et Combi "Time is everything" [188] pour les systèmes d'information cliniques. Les domaines d'applications nécessitant une prise en compte de l'information temporelle ne se réduisent pas aux seuls patients en USI, mais comprennent également l'organisation d'un parcours de soins, la gestion des traitements de malades chroniques comme les diabétiques [21]. Les raisonnements temporels dans les applications médicales posent autant de problème qu'ils sont nécessaires. Les directions de recherche pour résoudre ces problèmes qui ont été proposées principalement dans les années 90 sont multiples [59, 186] et sont toujours poursuivies [2, 11].

Nous revenons ici sur des techniques qui ont été proposées pour améliorer le monitoring de patients en USI pour lesquels les données temporelles sont les flux de données fournis par l'instrumentation des patients et les raisonnements doivent permettre de réaliser les tâches du monitoring : décision sur l'état du patient, information à l'équipe médicale et initialisation des actions curatives.

Ces méthodes se différencient par la complexité des modèles qui sont utilisés. Les méthodes de traitement de signaux [5, 57, 191] permettent d’apporter des solutions pour tirer une meilleure information des signaux biomédicaux en utilisant peu de connaissances *a priori* et donc des modèles sommaires. Les techniques d’intelligence artificielle sont également citées [27, 51] comme pouvant amener des améliorations dans la prise de décision en apportant des modèles plus complexes (*e.g.* modèles flous [206]). Ces modèles nécessitent beaucoup de connaissance *a priori* pour pouvoir les formaliser. L’extraction de tendances dans les signaux est une approche intermédiaire nécessitant assez peu de connaissances *a priori* et utilise des techniques d’intelligence artificielle pour tenir compte des contraintes temporelles.

B.3.1 Systèmes actuels

On a vu que les systèmes actuels de déclenchement d’alarmes en unité de soins intensifs utilisent des méthodes de traitement de signaux qui sont très basiques : une alarme est déclenchée lors d’un dépassement de seuil fixé (minimum et/ou maximum). Tsien [210] présente quatre algorithmes types :

- Moyenne sur une fenêtre flottante : L’algorithme prend en paramètre un nombre N valeurs précédentes (Le choix du nombre N dépend de la fréquence d’échantillonnage et du paramètre suivi). La moyenne de ces valeurs est comparée aux seuils pour déclencher des alarmes.
- Médiane sur une fenêtre flottante : De même que précédemment, N de valeurs précédentes sont considérées. Pour cet algorithme, on calcule la médiane plutôt que la moyenne des valeurs.
- Algorithme à délais : Une alarme est déclenchée uniquement lorsque N valeurs précédentes dépassent le seuil fixé.
- Échantillonnage des tests : Un paramètre N fixe le taux d’échantillonnage de valeurs à tester. Une valeur est alors utilisée toutes les N valeurs produites par le système pour être testées avec les seuils.

Pour tous ces algorithmes le bénéfice recherché est la robustesse aux artefacts. Les algorithmes permettent de ne pas déclencher les alarmes au moindre dépassement artefactuel du signal tout en conservant une sensibilité importante (montré par les évaluations de Tsien [210]). On effectue ainsi une sélection de fausses alarmes liées aux artefacts des signaux.

Pour tous ces algorithmes le déclenchement d’une alarme est lié à deux paramètres : le seuil qui a été fixé et le paramètre N qui correspond pour chacun de ces algorithmes à la taille de la fenêtre temporelle qui est considérée. Tsien montre également que ce second paramètre a une influence importante sur la qualité qui peut être attendue de l’élimination des fausses alarmes. Comme indiqué par Calvelo et *al.* [46], si cette taille est trop petite le bruit sera mis en évidence par les algorithmes, et si elle est trop large seules les évolutions à long terme seront considérées en omettant de potentiels variations significatives. Cette taille est liée à une échelle temporelle du paramètre suivi. Sharshar et *al.* [52, 192] proposent une méthode statistique pour caractériser ces tailles de fenêtre à partir des signaux en effectuant une analyse de la dynamique des paramètres physiologiques.

L’inconvénient de ces algorithmes est de ne s’intéresser qu’à une valeur unique qui caractérise un état du système sans prendre en compte sa dynamique. La tendance d’évolution d’un paramètre (croissant, décroissant ou stable) est une information plus riche qui peut être extraite à partir des signaux.

B.3.2 Extraction de tendances

L’objectif des méthodes d’extraction de tendances est de proposer une représentation synthétique d’un comportement sur un intervalle. Cela permet de segmenter le signal en zones de comportements homogènes. Les tendances qui sont recherchées généralement associées à des valeurs symboliques (Même si la pente de la courbe n’est pas utilisée. Elle pourrait être une information complémentaire.). Le problème de l’abstraction en tendances a été traité selon différentes perspectives [13, 20, 106, 107, 140, 158, 179, 211, 212].

Par le calcul des tendances (calcul qui peut être vu comme un pré-traitement pour réaliser la décision de l’état du système), on peut obtenir des alarmes plus significatives [184].

L’approche de Charbonnier et *al.* [57] pour calculer les tendances d’un signal dans un flux de données à partir de la technique CUSUM. La figure 1.4 illustre le résultat de l’algorithme : Des segments de droites de la forme $y(t) = p_i(t - t_{oi}) + y_{oi}$. L’agrégation progressive de données à un segment en cours de construction permet de cumuler les erreurs produites par l’approximation linéaire, et lorsque celle-ci dépasse un seuil

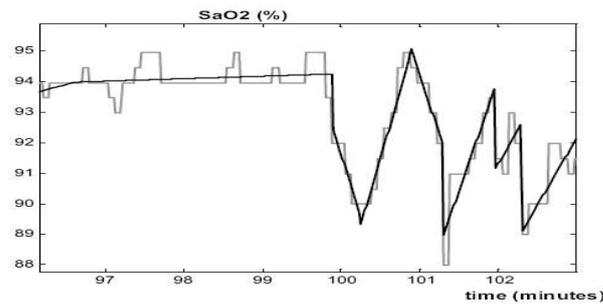


FIG. 1.4. Illustration de calcul de tendances par la méthode CUSUM (Extrait de [57])

acceptable, un nouveau segment débute. Un seuil supérieur permet de gérer les situations spécifiques d'un segment aux données anormales et les artefacts sont gérés lorsque deux valeurs consécutives dépassent un autre seuil à fixer. Cette méthode fonctionne avec des données peu bruitées mais à l'avantage de ne pas fixer de taille caractéristique pour construire les tendances : celles-ci sont calculées automatiquement et en ligne.

Dans Calvelo et *al.* [46], une étape de calibration est préliminairement nécessaire pour extraire la taille caractéristique des fenêtres à partir des données. Cette calibration est réalisée à l'aide d'une analyse temps-fréquence. Dans un second temps, des indicateurs de tendance et de stabilité sont calculés par des méthodes de filtrage. Et finalement, une répartition du plan caractérisé par les indicateurs de tendance et de stabilité définit les zones des différentes tendances qui peuvent être extraites : instable, stable, croissant et décroissant.

Les deux méthodes de Charbonnier et Calvelo ont été confrontées dans [52]. Cette étude met en évidence le fait que ces deux méthodes permettent d'identifier sur une signal de SpO_2 de manière complémentaires : Calvelo et *al.* détectent les désaturations et Charbonnier et *al.* les débranchements probables du capteur. Ces deux événements à la trace assez similaire (une chute importante de la valeur de la SpO_2) conduisent à une situation critique dans le cas de la désaturation et non-urgente s'il s'agit d'un débranchement du capteur. Cette confusion est souvent la source de nombreuses fausses alarmes en USI. Les tendances sont donc une solution pour réduire cette source de fausses alarmes.

En mono-signal, les auteurs indiquent que l'extraction de tendances ne permet pas de reconnaître des événements plus complexes dans les séries temporelles.

B.3.3 Utilisation de modèles complexes d'évènements

La recherche d'évènements identifiables par leur trace dans les signaux physiologiques doit permettre d'améliorer la décision sur l'état du système et l'information au personnel soignant. Pour cela, il faut des modèles complexes des événements, *i.e.* des modèles très expressifs et spécifiques. Contrairement aux tendances pour lesquelles un segment ne peut être associé qu'à un nombre très limité de valeurs symboliques (*e.g.* croissant, stable ou décroissant), des modèles des motifs très expressifs et plus spécifiques doivent être capables de distinguer avec un même formalisme une grande quantité de valeurs symboliques. La spécificité d'un modèle consiste à indiquer qu'un modèle correspond à un nombre assez restreint de situations (*e.g.* un événement de "désaturation" est beaucoup spécifique qu'une "décroissance" du paramètre SpO_2).

La prise de décision est améliorée par la possibilité de distinguer un plus grand nombre d'évènements dans les séries. L'expressivité des événements étant plus riche et leur combinatoire plus grande, la prise de décision permettra de distinguer plus de situations distinctes et de mieux délimiter les situations critiques et celles qui ne le sont pas. Cette expressivité permet également d'envisager le système au delà du simple déclenchement des alarmes comme les approches précédentes. Les modèles plus complexes peuvent servir à détecter des événements normaux pour informer sur l'état en cours du patient ou de son équipement de monitoring. La richesse de l'information qui peut être fournie par le système améliore le temps de réaction du personnel soignant en réduisant leur propre travail d'interprétation préalable de l'alarme et en répartissant au mieux leurs interventions au sein de l'USI dans le temps.

La méthode de A.S. Silvent [195] permet de construire de nouvelles abstractions à partir des tendances

qui informent le clinicien sur l'état : normal, instable, normal haut, normal bas, anormal haut ou anormal bas d'un paramètre. Cette approche permet de proposer une information plus riche que les simples alarmes (normal ou anormal), et potentiellement de hiérarchiser les alarmes en indiquant de manière non critique les états normaux (haut ou bas) et en utilisant des alarmes sonores pour les états anormaux.

Dans le travail de Tsien [211], des modèles à base d'arbres de décision et de réseaux de neurones sont utilisés pour distinguer des événements artefactuels contre des événements non-artefactuels. Mais de tels formalismes pourraient représenter des modèles d'événements plus spécifiques. Les résultats obtenus avec ces deux modèles permettent d'améliorer la détection d'artefacts.

Quiniou et *al.* [173] utilisent également des réseaux de neurones pour extraire dans les ECG les formes élémentaires des ondes (Ondes P, complexes QRS) afin d'identifier, dans un second temps si l'ECG est normal ou bien est le signe d'une possible arythmie cardiaque.

Steimann et *al.* [206] proposent des modèles à base de logique floue. Un événement est associé à une évolution dans le temps d'un paramètre. Cette évolution est caractérisée par un modèle flou linéaire. Les limites supérieures et inférieures pour reconnaître parfaitement un événement sont données par des fonctions linéaires par morceaux et la probabilité de reconnaître cet événement décroît linéairement en fonction de l'éloignement de ces limites. La figure 1.5 illustre des exemples de modèles flous.

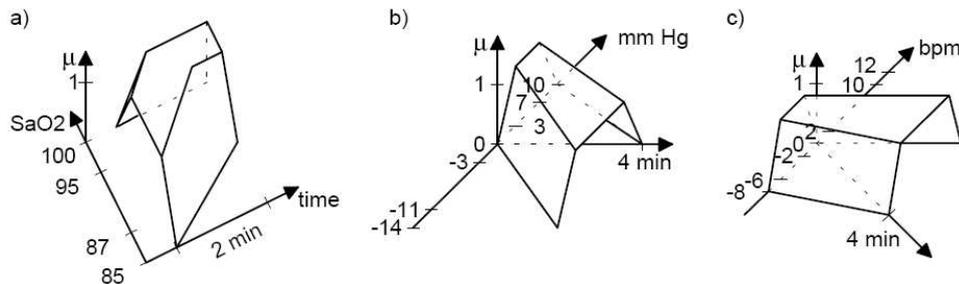


FIG. 1.5.

Exemples de tendances floues pour a) la saturation en O_2 , b) une chute de pression sanguine et c) des pulsations cardiaques croissantes. Les axes correspondent aux valeurs du paramètre, au temps et μ à la mesure de probabilité (Figure extraite de [206]).

B.3.4 Fusion multi-sources des informations

Dans [52], les auteurs concluent en affirmant qu'une détection multi-signaux permettrait d'extraire plus de modèles d'événements, mais la fusion d'information provenant de plusieurs sources d'informations est également une solution qui doit permettre de réduire les fausses alarmes liées à un dysfonctionnement d'un capteur. L'exemple du capteur de SpO_2 est de nouveau un bon exemple puisque sa déconnection accidentelle engendre une alarme (la déconnection étant confondue avec une désaturation du taux d'oxygène dans le sang). Cependant, on sait que physiologiquement la désaturation va s'accompagner de modification sur les paramètres physiologiques cardio-respiratoires (*e.g.* une oxymétrie basse s'accompagne généralement de signe de détresse respiratoire).

Dans [214], Tsien et *al.* utilisent deux modèles type obtenus par la construction d'arbres de décision et par régression logistique pour fusionner des informations sur trois signaux (Pulsation cardiaque, pression sanguine, et la pression partielle de CO_2). Les modèles permettent de détecter les artefacts dans les différents signaux. Les tests réalisés permettent de mettre en évidence l'amélioration apportée par la détection multi-sources.

B.3.5 Reconnaissance de scénarios

La reconnaissance de scénarios est à terme une solution pour améliorer efficacement les systèmes de monitoring. On entend par scénario un ensemble d'événements liés par des relations temporelles ou causales. Un scénario doit être le reflet des évolutions physiologiques typiquement observées sur un patient. Il permet de reconnaître des scènes typiques au plus tôt (la reconnaissance partielle d'un scénario permet d'anticiper l'envoi d'une alarme) et également de diminuer le taux de fausses alarmes (les scénarios servant

également à effectuer de la fusion d'information multi-sources) en permettant de croiser le scénario en cours de construction et les événements qui se produisent réellement. Les diagnostics obtenus tendent ainsi à être plus stables [205].

DiaMon-1 [206] intègre également la notion de scénarios sous la forme d'automates (finis déterministes) dans lesquels les états correspondent à des états physiologiques du patient et les transitions à des événements qui peuvent être détectés à l'aide de motifs flous à identifier dans le flux de données. L'expressivité d'un automate est pointée comme une limite à leur utilisation.

Dans [97], Fromont et *al.* utilisent la programmation logique inductive pour obtenir des représentations logiques de chroniques (ou scénarios) utilisées pour diagnostiquer des arythmies cardiaques à partir d'ECG (Electro-Cardiogrammes) et ABP (Pression sanguine artérielle). Ces représentations sont constituées d'événements (ondes P, complexes QRS,...) extraits des signaux par des réseaux de neurones. Les événements sont ensuite reliés par des relations temporelles *long*, *normal* ou *court* qui caractérisent les intervalles de temps (des connaissances expertes sont utilisées pour donner les plages de durées correspondantes aux valeurs *long*, *normal* ou *court*). En procédant ainsi, les relations entre les événements sont plus expressives que DiaMon-1.

La représentation des scénarios proposée par Dojat et *al.* [82] intègre la dimension temporelle quantitativement. Pour cela, les auteurs présentent le système *Déjà Vu* qui utilise des graphes temporels pour représenter les scénarios. Contrairement au système ATD [99] appliquant des reconnaissances de scénarios au diagnostic de l'hépatite B, la détection des scénarios est réalisée en ligne dans le flux de données.

Outre la possibilité de fusionner des informations et donc d'assurer une robustesse dans la détection en ligne des événements, l'utilisation de méthode de reconnaissance de scénarios permet d'envisager l'anticipation des alarmes. Lorsqu'un scénario est reconnu partiellement, il laisse présager de la suite de la scène.

B.3.6 Configuration automatique du monitoring

Aussi performant que soit un système de monitoring, la qualité de ses décisions dépend totalement de sa configuration. L'adaptation du système à la grande diversité des patients qui peuvent être rencontrés dans les USI et également à l'évolution d'un patient au cours de son traitement est un élément clé pour améliorer les solutions actuelles de monitoring.

F. Portet [171] propose une adaptation du système CALICOT, le système IP-CALICOT, qui inclut le pilotage des algorithmes de traitement des électrocardiogrammes (ECG) en sélectionnant les algorithmes à utiliser et leur configuration. À partir de ce pilotage, le système est capable de s'adapter au contexte d'utilisation du monitoring et d'identifier les signaux qui sont préférablement à utiliser pour détecter plus facilement des arythmies cardiaques sur des ECG bruités.

Parmi l'ensemble des voies d'amélioration qui ont été présentées, nous considérons que la détection d'événements complexes et de scénarios dans les séries temporelles est une piste très prometteuse pour l'amélioration des systèmes de monitoring de patients en USI.

C Construction de modèles pour le monitoring

Toutes les améliorations envisagées sont conditionnées à la disponibilité de connaissances formalisées, des modèles : la répartition du plan des indicateurs de tendance et de stabilité de Calvelo [47], les arbres de décision de Tsien [211], réseaux de neurones de Quiniou [173], Plus les modèles sont complexes et plus il est difficile de disposer de ces modèles. Il faut alors des moyens pour les construire.

Le problème de la solution de détection d'événements et de scénarios réside donc dans la possibilité de construire des modèles d'événements et de scénarios. Ceci pose deux questions différentes :

1. Quel formalisme permettra de capturer l'information significative dans les signaux pour identifier la trace d'un événement physiologique ?
2. Comment extraire un modèle (dans le formalisme adéquat choisi au préalable) correspondant à un événement pouvant être utile au monitoring ?

C.1 Formalismes pour les modèles d'évènements et de scénarios

Notre problématique fait intervenir quatre types d'objets : les faits réels, les observations, les modèles et les formalismes des modèles. La figure 1.6 met en relation ces différents objets. Tout d'abord, ce que nous appelons "faits réels" correspond à l'information idéale, *i.e.* continue et complète, de l'évolution du patient mais qui est inatteignable. Elle ne peut être que réduite par l'observation nécessairement finie et incomplète. Nous distinguons deux types d'observations :

- les observations cliniques réalisées par un clinicien,
- les traces du fonctionnement du patient obtenues par l'équipement technique du patient.

Les observations cliniques peuvent être mises en relation avec les traces enregistrées sur le patient.

Un modèle est une abstraction ou une formalisation représentant un ensemble d'observations. Une observation est ainsi une instance d'un modèle. Lorsque le modèle est obtenu à partir des observations d'un clinicien, on parle d'une formalisation des connaissances du clinicien, tandis que si ce modèle est obtenu à partir des traces d'un enregistrement, on parle d'une abstraction du modèle à partir des données. Finalement, les formalismes définissent la façon d'écrire un modèle. Un modèle est en quelque sorte une instance d'un formalisme.

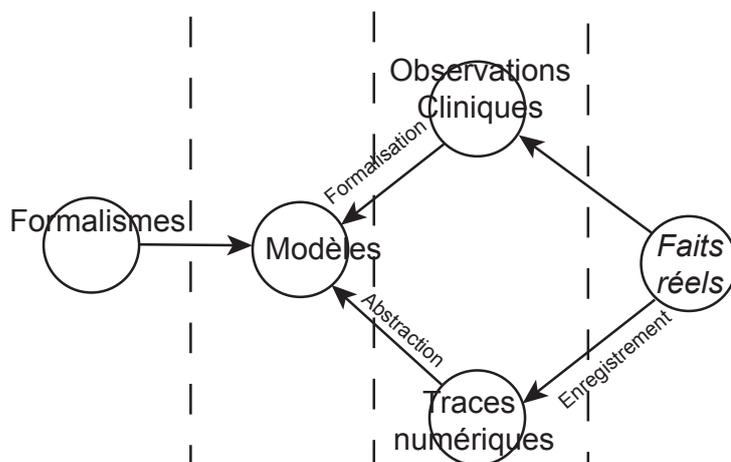


FIG. 1.6.
Agencement des Formalismes, Modèles, Observations et Données

Revenons plus particulièrement sur ces notions en les précisant pour les évènements et les scénarios.

Un *évènement* est une observation finie dans le temps qui peut être qualifiée médicalement (*e.g.* par la physiologie, par la thérapie, etc.). Cette qualification constitue le nom symbolique d'un évènement (correspondant à une signification médicale consensuelle par exemple la désaturation de l'oxygène du sang, le débranchement d'un capteur, l'introduction d'une drogue, ...). Un évènement est transitoire et dispose donc d'un support temporel fini (un instant ou un intervalle). Un *scénario* est une combinaison de plusieurs évènements que l'on peut observer chez un patient. On peut également associer un scénario à un nom symbolique (*e.g.* bradycardie, ...).

Chaque nom symbolique d'évènements (resp. de scénarios) est associé à un modèle d'évènements (resp. de scénarios). Par mesure de concision, on sera amené dans la suite à parler d'évènements (resp. de scénarios) à la place de modèles d'évènements (resp. modèles de scénarios).

Que se soit pour les modèles d'évènements ou de scénarios, les différents formalismes pour représenter les modèles ne sont pas équivalents. Le choix *a priori* de l'un d'eux est une question très difficile, car c'est de ce choix que dépend la possibilité d'identifier des modèles intéressants. En particulier, les distances entre les évènements et entre les scénarios sont très importantes (*cf.* section B.1.3 du chapitre 3) et elles sont dépendantes du formalisme qui aura été choisi.

Un modèle informatique est nécessairement calculable et peut alors être écrit dans un formalisme mathématique. Pour les modèles d'évènements, on peut envisager les formalismes des poids d'un réseau

de neurone, des chaînes de Markov ou encore des modèles flous de la forme d'un motif. Dans notre cas, nous avons utilisé deux formalismes : une description d'un évènement permettant de capturer la forme d'un segment d'une série temporelle et une description par un vecteur de caractéristiques quantitatives.

Pour les modèles de scénarios, parfois appelés chroniques dans la littérature, différents formalismes permettent de les représenter (graphes temporels, réseaux bayésiens ...). Un scénario est pour nous un ensemble d'évènements contraints par des relations temporelles quantitatives.

Nous reviendrons plus en détails sur les formalismes choisis dans la section B du chapitre 3.

C.2 Problématique de la construction des modèles

Une fois les formalismes définis, il faut effectivement construire des modèles applicables aux données de patients en unité de soins intensifs.

Notre approche de construction de modèles est très exploratoire. Il est donc nécessaire qu'elle soit réalisée hors-ligne, c'est-à-dire sur des données enregistrées au préalable dans des conditions réelles, mais qui ne nous contraignent pas pour les temps de calcul ou bien pour l'ordre d'arrivée des données. De plus, ceci ne limite en rien l'objectif d'amélioration des performances d'un système de monitoring. En effet, comme pour le système CALICOT [49], les modèles peuvent être construits hors ligne et ensuite utilisés dans le système de monitoring. La figure 1.7 illustre l'approche du système CALICOT qui sépare l'apprentissage hors ligne des modèles d'arythmies cardiaques, de l'utilisation de ces modèles pour la reconnaissance en ligne de ces mêmes arythmies.

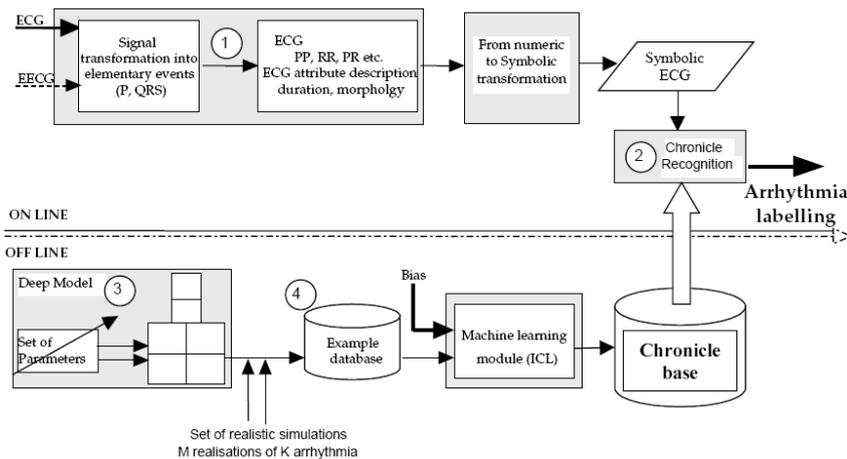


FIG. 1.7.

Architecture du système CALICOT [49] articulant un apprentissage hors-ligne de modèles et une utilisation en ligne de ces modèles.

Pour l'apprentissage de ces modèles, nous pouvons soit nous appuyer sur les connaissances des cliniciens (approche dirigée par la connaissance) soit explorer les données pour identifier au moyen de méthode de *data mining* les modèles les plus intéressants (approche dirigée par les données). Dans les deux cas, la tâche est difficile. Les cliniciens ne peuvent proposer facilement une formalisation de leur connaissance sous la forme de modèles informatiques. De l'autre côté, nous avons déjà pu montrer la difficulté de l'exploration automatique des données temporelles.

Les approches dirigées par les connaissances et dirigées par les données sont complémentaires. Le clinicien possède les connaissances cliniques qui permettent d'analyser les enregistrements de patients et la machine possède la puissance calculatoire qui va lui permettre de traiter un grand nombre de données. Si les deux approches n'ont pas permis jusque là de construire les modèles espérés, la combinaison des deux approches complémentaires peut apporter une solution à ce problème.

Nous nous intéressons donc à une solution qui permet à un clinicien de construire collaborativement avec une machine des scénarios et des évènements utilisables pour le monitoring de patients en unités de soins intensifs.

D Des systèmes pour aider à l’exploration de séries temporelles

Nous faisons dans cette section un état de l’art non exhaustif des outils informatiques existants pour aider les utilisateurs à la construction de modèles. Ces outils montrent trois différents types de “collaborations” dans l’exploration des séries temporelles. Tout d’abord, on présente des systèmes qui offrent des moyens de parcours des séries temporelles augmentés par rapport à un support papier. Ils permettent au clinicien de mieux appréhender les données et ainsi de faciliter la construction des modèles. Ensuite, une partie du travail de la construction des modèles peut être informatisée. On présente ensuite des outils qui permettent de construire à la demande de l’utilisateur des abstractions des données. Ces systèmes facilitent le traitement d’un grand nombre de cas au clinicien. Mais dans les deux cas, ces approches ne permettent pas une collaboration forte entre la machine et le système. Nous nous intéressons alors à une classe de systèmes qui sont centrés sur la construction d’annotations des séries temporelles en milieu médical. De tels systèmes apportent une solution de partage d’information intéressante pour construire une collaboration forte entre un clinicien et une machine. À la lueur de ces différents systèmes, nous serons en mesure de préciser l’axe de recherche de cette thèse pour l’exploration collaborative des séries temporelles.

D.1 Systèmes de parcours de séries temporelles

Les séries temporelles multivariées ne sont pas des objets facilement manipulables informatiquement. Un écran d’ordinateur est une contrainte imposée à la visualisation d’objets qui sur le papier peuvent être de grande taille (*e.g.* un sismogramme sur une bande papier). La visualisation d’un enregistrement est alors contrainte à réduire la résolution d’affichage de la courbe (en sous-échantillonnant le nombre de points à afficher) ce qui conduit à appauvrir l’information voire à la modifier (*e.g.* par effet de moiré). Il se pose également le problème de la *granularité* de l’information : toutes les informations temporelles ne méritent pas d’être résumées de la même manière. Ainsi, le système de visualisation de Bade et *al.* [18] permet une adaptation contextuelle de l’affichage. L’autre solution pour afficher l’information temporelle est de fragmenter le contenu et de focaliser l’attention de l’utilisateur sur seulement une partie d’un signal. L’utilisateur perd alors la vue globale de l’enregistrement. Il est donc nécessaire de mettre en place des outils qui permettent soit de représenter de manière différente les séries temporelles afin de donner une représentation correcte de l’information soit de les parcourir efficacement.

La dimension de visualisation de l’information temporelle des séries temporelles doit donc être prise en compte, mais ce n’est pas le centre d’intérêt de notre travail. De nombreux outils de visualisation de l’information temporelle ont été spécifiquement proposés pour prendre en compte les difficultés que nous avons succinctement mises en évidence. Dans [4], Aigner et *al.* dressent un panorama des méthodes de visualisation de l’information temporelle.

Nous nous intéressons dans ce paragraphe aux systèmes ou outils qui facilitent le parcours des séries temporelles en proposant des façons intuitives de retrouver de l’information dans les séries ou de représenter l’information temporelles différemment. L’objectif de ces outils est de faciliter une exploration des séries dirigées par l’utilisateur en lui donnant la capacité de percevoir rapidement l’information importante.

D.1.1 Interroger les séries temporelles intuitivement

La recherche d’une information est un élément primordial lorsqu’on veut manipuler des données en grande quantité. Cependant, pour des données complexes telles que les séries temporelles, la formulation de requêtes est difficile. Il faut pouvoir fournir des outils intuitifs de requête sur les séries temporelles. L’idée de QuerySketch [221] (*cf.* figure 1.8) est de permettre à un utilisateur de formuler une requête sous forme graphique pour retrouver parmi un grand nombre de courtes séries temporelles celles qui correspondent le mieux à la forme qui a été dessinée. La distance entre la forme et les exemples est calculée au moyen d’une distance Euclidienne (Voir la section B.2.1 du chapitre 3 pour plus de détails sur les distances entre séries temporelles). Cet outil est testé sur des données d’évolution journalière du montant d’enchères sur internet. Pour ces données, cette distance semble adaptée car toutes les séries sont de même taille et la correspondance point à point à également un sens.

QuerySketch est un outil simple et conçu pour faciliter une recherche de séries basée sur une forme.

L'avantage d'un tel outil est de répondre rapidement et correctement à une demande utilisateur.

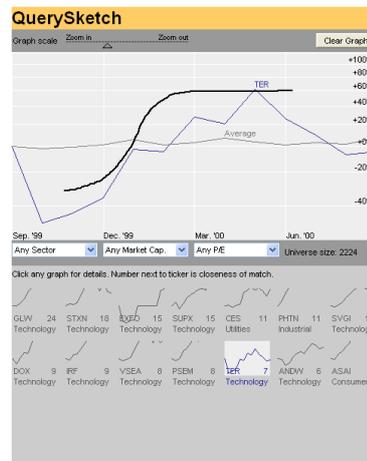


FIG. 1.8. Interface de QuerySketch. La zone supérieure permet de faire de requête graphique (en gras). Dans la zone inférieure de l'interface, le système donne les séries temporelles les plus similaires.

Le système TimeSearcher [44] propose également un outil de recherche de séries temporelles correspondant à des contraintes faites sous forme graphique, les *TimeBox*. Une *TimeBox* (en bleu sur la figure 1.9) définit une contrainte sur la valeurs des séries temporelles : la série temporelle doit “rentrer” par le côté gauche et “sortir” par le côté droit de la boîte. Lorsque plusieurs *TimeBox* sont utilisées les séries sélectionnées doivent vérifier la conjonction des contraintes ainsi définies. L'outil *TimeBox* semble mal adapté à l'exploration de longues séries temporelles. Par contre, il peut être utile pour mieux caractériser des ensembles de segments afin de construire la conjonction de contraintes strictes qui permet de les caractériser.

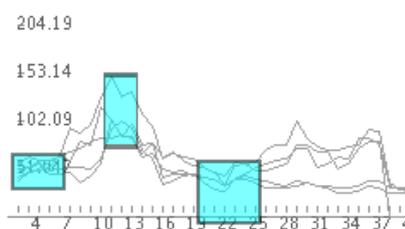


FIG. 1.9. Interface de TimeSearcher pour la réalisation de requête graphique sur des séries temporelles. Les *TimeBox*, rectangles bleus, définissent des contraintes à respecter pour les séries trouvées.

L'outil est surtout intéressant pour le dynamisme de son mode d'interaction. Il permet en effet de modifier les *TimeBox* et d'observer en même temps l'évolution des résultats de la sélection.

Dans ces deux outils, des efforts notables ont été faits pour permettre de réaliser des requêtes rapidement et de voir également très rapidement les résultats de la recherche. De la sorte, l'utilisateur peut réaliser de nombreuses requêtes et ainsi faciliter son appréhension des données au moyen de leur exploration.

D.1.2 Représenter l'information autrement

L'effort de visualisation des séries temporelles dans un espace de représentation également temporel fait face à des contraintes de conception d'IHM (Interface Homme-Machine). De plus, il se trouve que ce n'est pas nécessairement le bon mode de représentation des données. Certains systèmes proposent alors des visualisations alternatives des séries temporelles de sorte à donner accès à un autre point de vue, potentiellement meilleur pour des applications visées.

L'ordinateur peut être vu dans cette situation comme une "prothèse" sensitive [81]. Les séries temporelles ne sont perçues qu'au travers de leur dimension temporelle par un humain. Par exemple, lorsque nous observons une série temporelle complexe, nous ne "voyons" pas le spectre de Fourier de cette série. Pourtant, le spectre contient la même information que la représentation temporelle.

Représentation en arbre VizTree [139] est basé sur la méthode de représentation de motifs SAX [60]. SAX est une représentation symbolique de séries temporelles qui permet de transformer, à partir d'un alphabet *a priori*, une série numérique en série symbolique constituée de symboles de même taille. La méthode SAX s'accompagne de deux paramètres à fixer de manière *a priori* : le nombre de symboles de l'alphabet et la taille souhaitée des symboles. VizTree propose une façon plutôt intuitive et rapide d'explorer des motifs dans une série temporelle notamment en proposant une représentation de l'information sous la forme d'un arbre décomposant les motifs rencontrés dans la série. La figure 1.10 montre l'interface de l'outil.

L'outil propose de manipuler simultanément trois représentations des données :

- Une représentation du signal sur lequel peuvent être mis en évidence des segments correspondant au motif d'intérêt courant,
- Une représentation sous forme d'arbre qui permet de visualiser une "répartition" en nombre de l'ensemble des motifs possibles grâce à la représentation SAX paramétrée,
- Une représentation des segments d'intérêt cumulés sur un même graphe. Cette vue des segments permet de juger d'un coup d'oeil l'homogénéité de l'ensemble des segments sélectionnés.

Avec ces trois représentations des données, VizTree permet une visualisation résumant à la fois la structure locale et la structure globale des données.

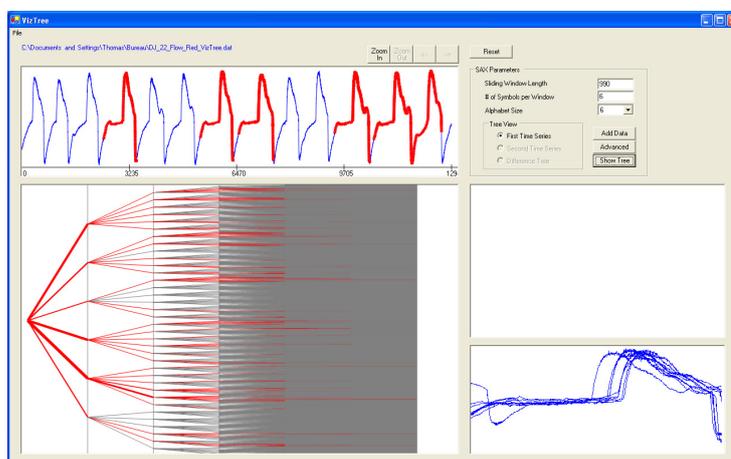


FIG. 1.10.

VizTree : Illustration de l'exploration d'un signal de débit respiratoire. En haut de la fenêtre on visualise la série explorée. En rouge sont les segments correspondant aux motifs retrouvés par le système à partir de la branche sélectionnée dans l'arbre de répartition des motifs dans la série, en bas à gauche de la fenêtre. En bas à droite de la fenêtre se trouve la superposition (ici homogène) des segments trouvés.

L'interface proposée par VizTree pour explorer les données est attrayante. L'utilisateur peut rapidement effectuer beaucoup d'expérimentations. Pour trouver des motifs intéressants dans les données ou se faire une idée de la représentativité de ceux-ci, l'utilisateur peut interagir sur la représentation temporelle ou sous forme d'arbre et observer les résultats sur les différentes représentations. Un grand nombre de motifs peuvent ainsi être explorés, mais à chaque nouveau motif exploré, les anciens motifs trouvés sont effacés ne permettant pas par exemple de comparer les répartitions des motifs ou les co-occurrences

éventuelles. VizTree dispose également d'un outil intéressant pour comparer deux séries temporelles sur la base de leur représentation en arbre en construisant un "arbre de différences".

Cependant, au delà d'une certaine taille d'alphabet, le temps de calcul est long et on perd tout l'intérêt du dynamisme de l'exploration qui est offerte. Par conséquent, seuls des motifs simples peuvent être traités. Une autre limite à l'exploration est la taille fixée des motifs recherchés (en nombre de symboles dans le motif et donc également en taille du motif complet). Par conséquent, deux motifs au même profil, mais dilatés dans le temps ne pourront être pas considérés comme "identiques". Ceci est clairement une limitation pour nos applications médicales dans lesquelles des segments identiques peuvent être dilatés l'un par rapport à l'autre.

Dans tous les cas, l'exploration des séries temporelles avec VizTree nécessite une bonne maîtrise de la représentation SAX pour paramétrer la transformation et comprendre la représentation en arbre.

Utilisation de graphes de dispersion Le système TSEplorer (*Time Series Explorer*) de Craig et al. [69] est un outil qui combine l'exploration sur une représentation temporelle de séries (données provenant de puces à ADN) et d'un graphe de dispersion (*scatterPlot*).

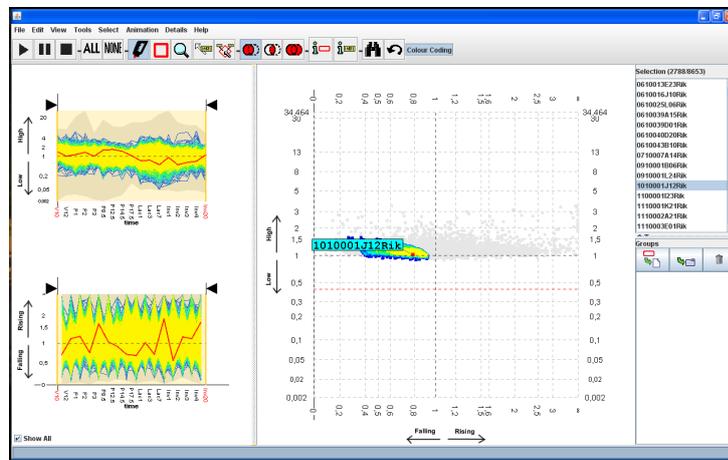


FIG. 1.11.

Interface de TSEplorer (Extraite de [69]). Sur la gauche, la superposition de l'ensemble des séries temporelles multivariées qui sont étudiées (2 séries par enregistrement). Au centre, le graphe de dispersion pour lequel chaque point correspond à une série temporelle. Les points en couleurs correspondent aux séries qui ont été surlignées en couleurs dans les graphes de gauche. Le point sélectionné correspond à la série en rouge dans l'interface de gauche.

Chaque série temporelle d'un ensemble de séries temporelles à étudier est représentée par deux points correspondants aux valeurs moyennes des paramètres considérés sur une portion des séries (Sur la figure 1.11, les points sont calculés sur les séries temporelles en entier). On peut sélectionner et observer les séries temporelles en même temps sur le graphe temporel et le graphe de dispersion.

Lorsqu'on sélectionne une durée courte sur laquelle calculer le graphe de dispersion, on obtient différents graphes en fonction de la position dans le temps de la fenêtre temporelle. TSEplorer permet d'animer le graphe de dispersion en montrant l'évolution du graphe de dispersion avec l'avancée de la fenêtre temporelle dans le temps. L'utilité d'un tel outil est de définir un sous-ensemble de séries temporelles qui ont initialement les mêmes caractéristiques sur le graphe de dispersion et de suivre au cours du temps leur évolution soit pour affiner la construction d'un groupe de séries similaires, soit pour identifier des caractéristiques communes dans l'évolution de ces caractéristiques.

TSEplorer est un outil original pour permettre de mettre en évidence des similarités entre des séries temporelles. Les données qui peuvent être traitées par cet outil sont assez contraintes puisqu'il faut des séries temporelles bi-variées qui font toutes la même longueur. La représentation sous forme de graphe de dispersion est ici intéressante pour illustrer un mode de parcours des séries temporelles qui bénéficie de plusieurs représentations des données.

Représentation textuelle des données Le système BabyTalk de Portet et al. [172] est également une approche originale pour faciliter l'appropriation de séries temporelles puisqu'il propose de construire un

texte qui résume le contenu de séries temporelles provenant de capteurs disposés sur un bébé surveillé en service de néonatalogie. Les études [135] sur lesquelles les auteurs s'appuient ont montré que l'information textuelle était, au final, plus pertinente que les signaux pour la prise de décision des cliniciens.

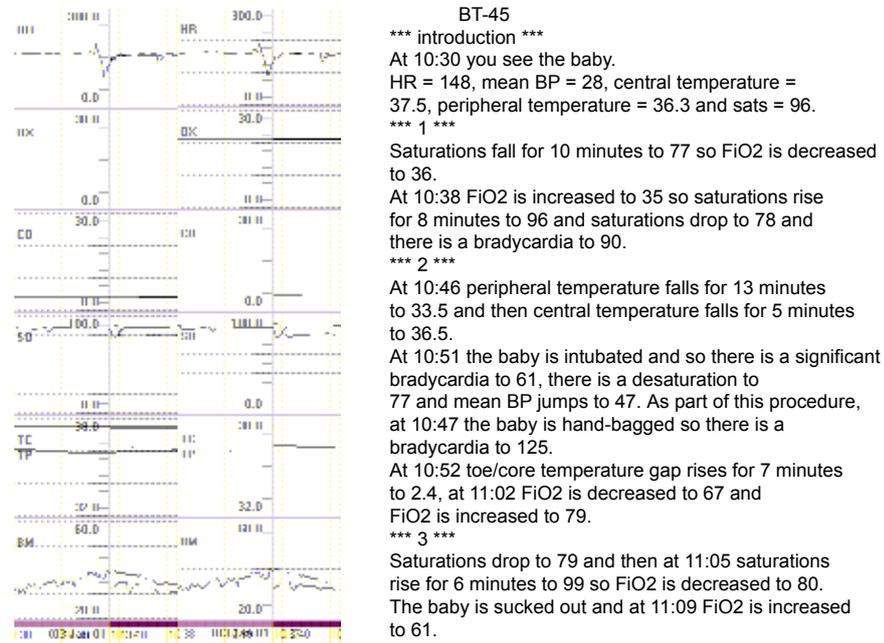


FIG. 1.12.

Sur la gauche : Données d'un bébé en néonatalogie résumées par BabyTalk. Sur la droite : Exemple de résultat de production textuelle utilisant la méthode BT-45 [135] à partir du signal de gauche.

Dans le système BabyTalk, il n'y a pas proprement dit d'exploration des données puisque le système a pour but l'aide à la décision des cliniciens. Pour cela, la traduction des signaux sous forme textuelle est réalisée sans intention de revenir sur les signaux. Cette approche de la représentation des séries temporelles nous apparaît cependant très intéressante. Dans le contexte médical où les séries temporelles représentent une masse d'information importante, la construction d'un résumé textuel est une solution pour aider les cliniciens à mieux les appréhender. Ils nécessitent pas l'apprentissage de la lecture d'une série temporelle et semblent plus proches des concepts qui sont enseignés dans le cadre d'un apprentissage théorique de la médecine.

L'intérêt des deux premiers systèmes (VizTree et TSExplorer) n'est pas simplement de donner accès à une autre représentation des données mais également de manipuler en parallèle les deux. De sorte que l'utilisateur ait une perception des données au travers de toutes ces représentations et que cela l'aide à les appréhender. Le système BabyTalk ne vise pas l'exploration des données, mais ouvre la possibilité originale d'utiliser une représentation textuelle pour faciliter la compréhension des données temporelles.

D.2 Système d'aide à l'exploration de séries temporelles

Par rapport aux systèmes précédents qui étaient des outils pour faciliter le parcours de séries temporelles, les outils présentés dans cette section intègrent des fonctionnalités plus évoluées qui permettent à l'utilisateur d'automatiser une partie de l'exploration. Précédemment, la construction des modèles restait à la charge de l'utilisateur, et la collaboration n'était effective que pour focaliser l'attention sur les données intéressantes dans la grande masse de données des séries temporelles. Les systèmes que nous présentons ici abordent la collaboration dans la construction des modèles puisque par leur calcul, ils vont aider le clinicien à formaliser les modèles des données qu'il désire modéliser.

TimeSearcher 2 et 3 sont des évolutions successives du système TimeSearcher et offrent des méthodes d'exploration des données plus riches à chaque version. Et KNAVE-II est un système qui est consacré à l'abstraction automatique des séries temporelles

D.2.1 TimeSearcher 2 et 3

TimeSearcher2 (cf. figure 1.13) est un logiciel interactif d’exploration d’une base de séries temporelles multivariées utilisant des outils d’interrogation par des exemples (*query-by-example*) : les *TimeBox* et les *SearchBox*. Les *TimeBox* ont été présentées avec le système TimeSearcher dans la section précédente.

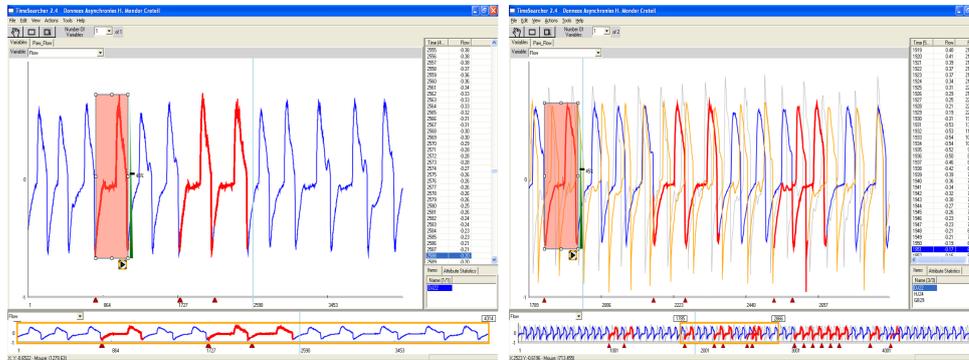


FIG. 1.13. TimeSearcher2 : utilisation des *SearchBox* pour la recherche de motifs. La boîte rouge indique le pattern recherché. Dans le bas de cette fenêtre, l'utilisateur peut visualiser les segments ainsi détectés et surlignés en rouge.

Par rapport à la première version du logiciel, un outil interactif nommé *SearchBox* a été ajouté. Les *SearchBox* permettent à un utilisateur de sélectionner un segment de séries temporelles et de rechercher dans l'ensemble des séries de la base les motifs similaires. Un contrôle permet à l'utilisateur d'ajuster le seuil de similarité entre les segments qui sont comparés à l'aide d'une distance Euclidienne. De part l'utilisation d'une distance Euclidienne, le système ne permet cependant pas de tenir compte de l'hétérogénéité des motifs. Ceci limite sa capacité de détection de motifs similaires pour des données qui comportent une grande variabilité de motifs (la figure 1.13 montre des tests réalisés avec l'outil sur nos propres données pour lesquels deux motifs *similaires* peuvent être très différents selon la distance Euclidienne).

TimeSearcher3 [45] est la version suivante de l'outil de Bueno et al. pour l'exploration de séries temporelles. TimeSearcher3 ajoute la dimension de prédiction de l'évolution des séries temporelles à son panel d'outils d'exploration. L'objectif étant non pas de faire de la prédiction de séries temporelles sur la base de modèles (comme les modèles économétriques, tel que le modèle ARMA, peuvent proposer), mais de faire des prédictions sur la base d'exemples similaires qui sont disponibles dans la base de données.

Lorsqu'un début de série temporelle est à disposition, e.g. les premiers jours d'une enchère sur internet, on cherche à savoir comment va évoluer cette série et quel sera le montant de l'enchère finale. Pour cela, la portion de série disponible est comparée aux données de la base et le système affiche une évaluation probabiliste de l'évolution (sous la forme de "river plot", cf. figure 1.14).

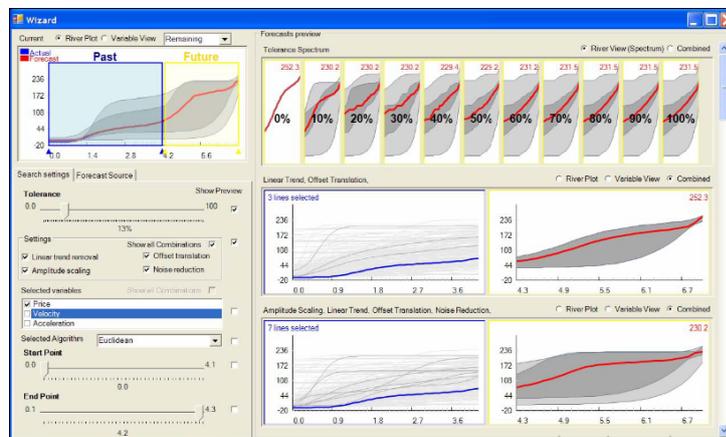


FIG. 1.14. Interface de TimeSearcher3 pour l'exploration d'une prédiction (Figure Extraite de [45])

Sur la figure 1.14, l'interface d'exploration d'une prédiction permet à l'utilisateur d'interactivement modifier les paramètres de similarité de la portion de série avec les exemples de la série temporelle et de voir simultanément le résultat avec plusieurs paramètres.

La même démarche d'exploration des séries temporelles que les versions précédentes est utilisée pour la prédiction : Les retours faits à l'utilisateur sont rapides facilitant l'exploration. Les explorations peuvent se faire selon plusieurs modes et il est mis l'accent sur un affichage dynamique pendant la manipulation.

Le système TimeSearcher3 est donc un système qui permet une exploration interactive des données grâce à une multitude d'outils intuitifs : *TimeBox*, *SearchBox* et calculs de prévisions. Ces calculs de prévision rendent l'outil TimeSearcher3 plus proche d'une exploration collaborative des données que d'un parcours comme dans les deux versions précédentes. La machine n'est plus simplement un facilitateur pour l'utilisateur, mais un organe de traitement des données qui permet d'augmenter les capacités de compréhension des données de l'utilisateur.

D.2.2 KNAVE-II

KNAVE-II [30, 144, 187] est un outil de traitement intelligent de séries temporelles biomédicales à base de connaissances. Sur la figure 1.15, on voit que l'interface permet de visualiser les séries temporelles ainsi que leur abstraction. Les abstractions temporelles sont réalisées par le système à partir du *moteur* d'abstraction IDAN [30]. Ces abstractions peuvent être gérées de manière automatique grâce à l'utilisation d'une base de connaissances qui permet de retenir pour les différents types de signaux comment ceux-ci *doivent* être abstraits. Plus qu'une simple base de connaissances, le système est en fait couplé avec une ontologie (dont une partie est visible sur le panneau de gauche de la figure 1.15), incluant par exemple des relations entre les données et leurs abstractions.

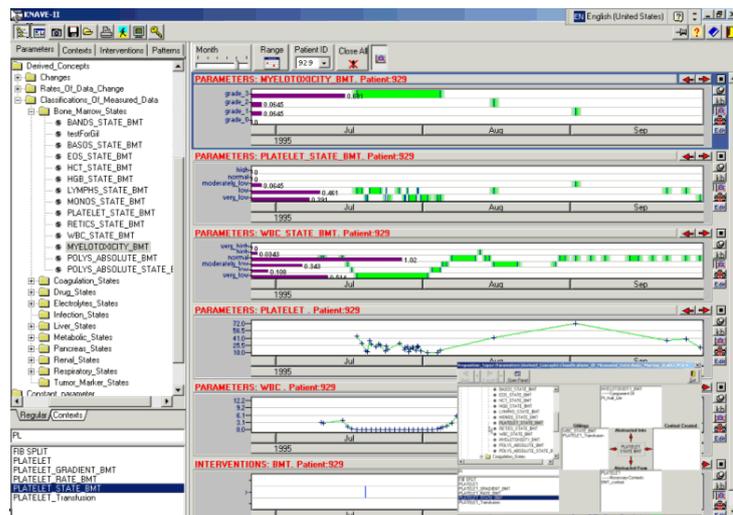


FIG. 1.15.
Interface de KNAVE-II (Figure Extraite de [189])

KNAVE-II est surtout un système qui s'attache à centraliser autour de l'outil d'abstraction temporelle des fonctionnalités de visualisation (comme des *scatterPlot*, des cartes de Kohonen ou des méthodes pouvant prendre en compte la granularité des données), des méthodes de pré-traitements, des méthodes de recherche, de parcours interactifs et des méthodes statistiques qui peuvent aider les cliniciens à réaliser des études à partir des données temporelles.

En proposant ainsi un foisonnement d'outils différents dans un même système, KNAVE-II est un outil d'exploration des données qui peut aider l'utilisateur à toutes les étapes d'une construction de modèles à partir de séries temporelles. Du point de vue de la collaboration, ce système reste un outil dans le sens où il reste dirigé entièrement par l'utilisateur.

En résumé, les systèmes d'aide à l'exploration de séries temporelles cherchent à mettre à disposition un ensemble varié d'outils qui vont permettre interactivement d'enrichir la connaissance des données en

utilisant les capacités de calcul de la machine (ou d'un réseau de machines comme peut le proposer l'outil TSNet [119]). Ils ont chacun intégré des fonctionnalités riches et variées pour permettent à l'utilisateur de construire sa propre exploration des données.

Dans ces systèmes, une collaboration apparaît pour réaliser les tâches de recherche de l'information significative dans les données comme le proposait les systèmes du paragraphe précédent, mais en ce qui concerne la construction des modèles, les systèmes ont une collaboration limitée. Les modèles ne sont pas construits par le système. C'est plutôt l'utilisateur qui formalise à l'aide de la machine comme outil de calcul. Pour la construction de modèles, les communications entre le système et l'utilisateur peuvent se schématiser par des interactions du type requête/demande. Il s'agit là d'une collaboration déséquilibrée.

Dans une perspective de la construction d'une collaboration qui permet de soutenir la création et la découverte de connaissance, il nous semble que d'une part le système et l'utilisateur doivent partager le même but à accomplir et que pour le résoudre, ils doivent communiquer par un flux symétrique de données. Mais ces données doivent être comprises mutuellement, ce qui n'est pas nécessairement évident entre une machine et un humain. Les annotations de séries temporelles, déjà utilisées pour communiquer entre cliniciens, sont des entités qui peuvent être comprises à la fois par la machine et par un clinicien. C'est pourquoi nous nous intéressons aux systèmes qui permettent aux cliniciens de réaliser des annotations.

D.3 Systèmes pour l'annotation de séries temporelles médicales

Au delà de notre approche de la collaboration homme/machine, dans laquelle l'annotation peut servir de support à l'interaction entre le système et l'utilisateur, l'annotation des séries temporelles est un problème en soi dans les applications médicales.

Dans l'objectif de l'amélioration des systèmes de monitoring, il est nécessaire de disposer d'une grande masse de séries temporelles biomédicales de qualité. La qualité de ces données passe par la mise à disposition d'annotations. Les annotations sont en effet une façon de partager les informations sur les données et collectivement d'enrichir les données brutes de connaissances utiles pour les comprendre. L'annotation est alors un support pour la collaboration et le partage des données.

Les bases de données de signaux biomédicaux annotés permettent la mise en place de solutions pour leur exploration approfondie. Des avancées peuvent ainsi être réalisées dans l'identification des événements et des scénarios recherchés pour la conception d'outils performants de monitoring. En conséquence, il devient nécessaire de disposer d'outils qui permettent de les annoter facilement et rapidement.

En présentant ici le système MIMIC-II (*Multi-parameter Intelligent Monitoring for Intensive Care*) nous souhaitons mettre en évidence d'une part la difficulté que peut représenter la tâche d'annotation et l'utilité pratique qui peut être tirée d'un outil capable d'automatiser cette tâche. D'autres communautés telles que la communauté d'exploration des informations des chaînes d'ADN rencontrent également le problème de l'annotation de données en grande quantité [77].

Le système MIMIC-II [1, 178] est une station d'annotation associée à une imposante base de données de signaux biomédicaux. L'architecture permet d'intégrer des données, des modèles et des méthodes de raisonnement sur des données de soins intensifs. L'objectif est de proposer un banc de tests pour aider à la recherche sur le monitoring intelligent de patients en unités de soins intensifs.

Le système propose une gamme assez complète d'outils pour exploiter les données acquises :

- méthodes d'anonymisation des données,
- outils d'annotations *a posteriori* des enregistrements, des diagnostics, des thérapeutiques et des alarmes enregistrées,
- méthodes de monitoring avancés (Pré-traitements, reconnaissances de motifs, raisonnements physiologiques, identification de modèles)

Le système MIMIC-II est fusionné avec le système MAITA (*Monitoring, Analysis, and Interpretation Tool Arsenal*). Cet outil, également développé au sein du MIT par Doyle et al. [84], propose une architecture pour construire des systèmes de monitoring intelligents avec des traitements distribués des données.

L'orientation choisie pour faciliter l'annotation de signaux biomédicaux n'est pas celle de l'automatisation, mais celle de la mise à disposition d'interfaces riches pour accélérer l'annotation en permettant une visualisation d'un cas clinique globalement. Le projet MIMIC-II a permis de développer une station intégralement dédiée à l'annotation des signaux. Cette station dispose d'une interface répartie sur 4 écrans (*cf.* figure 1.16) qui permettent à l'annotateur de visualiser une grande masse d'informations utiles en

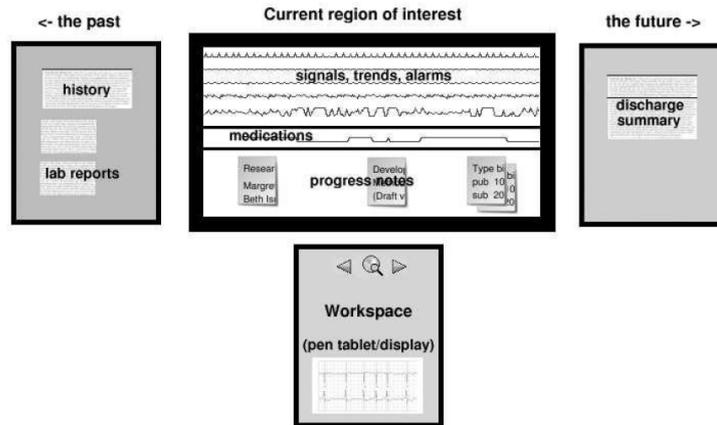


FIG. 1.16.

Organisation du poste d'annotation de MIMIC. Le poste utilise une interface à 4 écrans qui permet à l'annotateur d'avoir une vision globale de l'historique du patient. (Figure extraite de [143]).

même temps. Les écrans sont logiquement disposés pour donner l'intuition de l'historique du patient :

- L'écran de gauche résume les connaissances acquises dans le passé, c'est-à-dire les résultats des examens médicaux qui ont été réalisés.
- L'écran de droite permet d'afficher la prospective qui a été ou est en train d'être construite. L'écran de droite est ainsi associé au futur du patient.
- L'écran central supérieur est consacré à l'affichage des séries temporelles et leurs annotations, c'est-à-dire le présent.
- L'écran central inférieur permet la navigation dans un enregistrement.

MIMIC-II butte sur l'acquisition plus rapide d'annotations complètes, riches et de qualité pour l'immense quantité de données que représentent les séries déjà à disposition. Même si la station d'annotation proposée par MIMIC-II correspond à une solution très intéressante qui permet d'obtenir des annotations riches, le processus reste encore très long et fastidieux pour l'annotateur. La solution est clairement de profiter de méthodes automatiques de construction des annotations des séries temporelles.

D.4 Positionnement de notre proposition

Le monitoring est une tâche complexe qui implique d'avoir des systèmes capables d'identifier la dynamique d'un système. Les systèmes actuels sont perfectibles et leur amélioration est nécessaire. Leur amélioration passe par la détection dans les signaux physiologiques d'évènements complexes riches de sens et de scénarios. Cependant, il n'existe ni évènements ni scénarios suffisamment formalisés dans un cadre générique de monitoring de patients d'USI. Il faut donc pouvoir proposer des méthodes pour définir ces évènements et bâtir les scénarios utiles à reconnaître.

Notre proposition concerne un outil d'aide à l'interprétation de séries temporelles. La première série de systèmes présentés plaidait pour la mise en place de modes d'interaction qui permettent de parcourir les séries temporelles de manière dynamique et pouvant aider à la compréhension des données. En particulier, la mise à disposition de représentations alternatives des données permet d'augmenter la capacité de l'utilisateur à identifier des corrélations ou des caractéristiques intéressantes des données. On peut indiquer qu'elles sont essentiellement consacrées à des applications dans lesquelles les séries temporelles sont de petite taille. La seconde série de systèmes a mis en évidence qu'un système d'exploration de séries temporelles doit intégrer et mettre à disposition facilement un grand nombre d'outils (*e.g.* pré-traitements, abstractions, statistiques) qui vont permettre à l'utilisateur de mener à bien son exploration et que celui-ci doit pouvoir agencer facilement ces éléments pour se construire son propre parcours exploratoire.

Ces outils restent des outils à la disposition des cliniciens. Nous proposons de construire une collaboration forte entre la machine et le clinicien de sorte à ce qu'ils s'enrichissent mutuellement de leurs compétences spécifiques. La notion d'annotation a attiré notre attention pour permettre un échange de connaissance entre la machine et un clinicien. De plus, la construction de modèles formalisés passent selon nous par une compréhension plus globale des séries temporelles. L'annotation est un premier pas qui

permet d'apporter du sens aux enregistrements. En allant plus loin, les systèmes informatiques doivent aider les cliniciens à *interpréter* les enregistrements des patients.

Notre proposition vise à construire un système pour l'interprétation collaborative d'enregistrements de patients d'USI.

Chapitre 2

Interprétation collaborative de séries temporelles

Pour remplir l'objectif pratique de construction de modèles d'évènements et de scénarios, nous proposons de réaliser une interprétation collaborative des séries temporelles multivariées.

Une interprétation est une tâche d'élucidation du sens. Par un parcours interprétatif entre des considérations locales et globales des données, l'interpréteur recrée de manière subjective le sens de sorte à le mettre en cohérence avec ses connaissances actuelles. L'interprétation des séries temporelles doit permettre de faire émerger une compréhension des séries temporelles exemples à partir de laquelle des modèles d'évènements et de scénarios pourront être construits ou affinés.

La collaboration entre deux agents, un clinicien et un système informatique, doit permettre de réaliser l'interprétation en dépassant les difficultés que chacun rencontre individuellement. L'interprétation étant subjective, nous proposons une approche de la collaboration fondée sur un couplage structurel entre les deux agents. Ce couplage inspiré du paradigme de l'Enaction de Varela doit permettre de faire collaborer les deux agents en conservant leur autonomie d'interprétation. La communication est alors assurée par l'échange d'annotations.

L'interprétation réalisée par le système informatique et l'utilisateur implique que les deux agents formalisent des modèles. En leur faisant réaliser une interprétation on remplit donc les objectifs de formalisation des données par le clinicien et de construction de modèles informatiques d'évènements et de scénarios.

La tâche d'interprétation est au centre d'un axe opposant les approches dirigées par les données et celles dirigées par les connaissances. Une approche dirigée par les données permet à un système automatique d'extraire des connaissances sous la forme de modèles à partir des séries temporelles que nous avons à disposition. Tandis qu'une approche dirigée par les connaissances d'un expert tâche d'interroger les connaissances pratiques d'un clinicien pour formaliser des modèles. L'interprétation cherche à faire les deux à la fois : un clinicien qui interprète des données va utiliser à la fois ses connaissances et les données pour se construire une compréhension de leur ensemble. En particulier, réaliser une tâche d'interprétation implique la construction de modèles. L'objectif de la construction de modèles n'est donc pas explicite, mais est réalisé implicitement dans l'interprétation.

La collaboration est également le pivot entre une solution qui consiste à faire construire les modèles par une machine et celle qui laisse le clinicien faire cette modélisation. L'utilisation d'une machine permet de réaliser des apprentissages automatiques à partir d'un grand nombre de données pour en extraire des modèles qui sont statistiquement intéressants. Tandis que les modèles qui sont extraits par la modélisation d'un clinicien s'appuient plus sur des cas particuliers spécifiques et tiennent compte d'une information contextuelle qui n'est pas accessible à la machine. Réaliser une collaboration entre une machine et un clinicien doit permettre de bénéficier des avantages des deux approches.

Il est potentiellement plus difficile de réaliser une interprétation de séries temporelles que de faire uniquement une extraction de connaissances à partir des celles-ci ou des connaissances d'un clinicien. Mais les approches prises indépendamment n'étant pas suffisantes pour réaliser notre tâche de construction de scénarios et de modèles d'évènements, nous ambitionnons de résoudre ce problème en les englobant. Nous proposons donc la réalisation d'une interprétation collaborative des enregistrements de patients.

Dans ce chapitre, nous introduisons la notion d'interprétation et en particulier l'interprétation de séries temporelles multivariées. Ceci nous permettra d'opérationnaliser cette notion complexe dans le but de faire réaliser une "interprétation" pour des séries temporelles par une machine qui pourra ainsi construire les modèles formels attendus. Nous présentons ensuite notre approche de la collaboration homme/machine basée sur le paradigme de l'Enaction de F. Varela. Ce paradigme nous donnera également un éclairage sur la notion d'interprétation réalisée collaborativement.

A Interprétation

A.1 Caractéristiques d'une tâche d'interprétation

L'interprétation peut prendre différentes formes, en particulier elle ne se limite pas à la seule interprétation de écrits (herméneutique). L'interprétation peut être artistique, traduction, juridique ou même l'interprétation de "notre présence au monde" qui est dite philologique [104]. L'interprète d'une œuvre d'art (acteur, chanteur, danseur,...) doit rendre une œuvre déjà construite aussi *juste* que possible, c'est-à-dire à la fois proche de l'objet et du public destinataire du message. L'interprétation de la loi est une mise en pratique des textes pour statuer sur une situation particulière. Pour une traduction, on peut avoir besoin d'un interprète qui extrait le sens d'une langue et d'une culture étrangère pour la transmettre dans une autre dans le but d'en assurer la compréhension. Enfin, l'interprétation de "notre présence au monde" interroge sur la thèse selon laquelle "il n'y a pas de fait seulement des interprétations"¹. Cette forme de l'interprétation est traitée par les philosophes et s'inscrit dans le fonctionnement de la nature humaine. Pour l'homme, l'interprétation est constamment à l'œuvre sans même qu'il en soit conscient.

W. Iser [122] dresse un panorama de l'interprétation qui ajoute à cette liste l'interprétation de l'Histoire et des Hommes (par la psychologie et l'ethnologie). Mais cette liste n'est certainement pas exhaustive.

Ces formes d'interprétation ont à la fois tout et rien à voir avec la notion d'interprétation des séries temporelles comme trace de fonctionnement d'un système dynamique. Il n'y a aucune œuvre à jouer dans l'interprétation de séries temporelles, mais nous partageons l'idée de la recreation du sens à partir de l'objet initial : la partition ou le texte de l'artiste et les données enregistrées pour notre tâche. Si nous nous intéressons à comprendre des situations particulières au regard de connaissances médicales comme le fait le juriste au regard de la loi, ce n'est pas en premier lieu les connaissances que nous interprétons au regard de la situation comme le propose la démarche du juriste, mais plus l'inverse. Les interprétations

¹Nietzsche, La volonté de puissance, n°481.

desquelles nous pouvons nous sentir le plus proche sont certainement les interprétations des écrits ou bien l'interprétation comme traduction. Dans les deux cas, il peut être mis en avant la recherche d'une compréhension d'objets à disposition.

De toutes ces différentes facettes, il faut identifier les points clefs qui constituent l'essence même de la notion d'interprétation.

Selon I. Weiss [222], l'interprétation apparaît d'abord comme la mise en rapport de deux instances : un objet qui ne livre pas d'emblée sa signification, qui se présente bien comme signifiant mais qui ne met pas explicitement et spontanément cette signification à découvert, et ce que on peut appeler un sujet, qui s'attelle à la tâche, plus ou moins savante, plus ou moins technique, de l'élucidation du sens. La complexité de la tâche provient de la distance plus ou moins large, mais irréductible qui sépare l'objet et le sujet [98].

W. Iser [122] conclut de son étude de différentes interprétations que c'est une performance qui implique un engagement de l'interpréteur plutôt qu'une simple validation, et qu'elle produit une adaptation plutôt qu'une explication. L'objectif identifié est semblable à celui de I. Weiss, c'est-à-dire de réduire un espace liminal entre l'objet et le sujet (le sujet est réduit à un *registre*).

Ces différents travaux nous amènent à utiliser un point de vue sur l'interprétation à l'aide de trois caractéristiques :

- l'interprétation en tant qu'acte de compréhension est le résultat d'un parcours non-linéaire s'appuyant sur des dualités entre local et global ;
- l'interprétation est une appropriation individuelle par la recréation de l'objet par le sujet, et par ce fait, elle lui est propre et non directement partageable, échangeable ;
- l'interprétation est une œuvre sans fin qui cherche à rapprocher l'objet, le sujet et son contexte par la construction d'une cohérence des trois ;

Dans les trois paragraphes suivants, nous revenons sur chacune de ces caractéristiques.

A.1.1 Cercle herméneutique

En premier lieu, l'interprétation moderne n'est plus vue comme un dogme, intangible et immuable, défendu alors par une autorité (*e.g.* une autorité religieuse lorsqu'il s'agissait d'interpréter les textes bibliques), mais plutôt comme un chemin non linéaire que Schleiermacher [183] présente sous le terme de "cercle herméneutique".

Selon Schleiermacher, le langage est encodé à la fois par la grammaire (son usage des signes) et la psychologie de son auteur. Le langage étant nécessaire à la pensée et la pensée nécessaire au langage, les deux notions sont duales.

La démarche herméneutique de Schleiermacher s'appuie donc sur cette dualité en proposant une exploration parallèle de chacune des deux facettes. Partant de cette définition, une interprétation grammaticale du langage permet d'extraire l'usage des signes propre à l'auteur, et l'interprétation psychologique permet d'extraire de l'usage stylistique des signes les traits psychologiques de l'auteur.

En approfondissant la réflexion, Schleiermacher décompose l'interprétation grammaticale en un nouveau couple détermination/exclusion et l'interprétation psychologique en un couple comparaison/divination. La détermination provient du fait que le sens donné aux mots d'un texte dépend de son entour et duale de ce qu'il en exclut. Le couple comparaison/divination met en parallèle la compréhension de l'auteur comme individualité (divination) et l'auteur comme appartenant à un groupe (comparaison) par lequel ses discours doivent être interprétés.



FIG. 2.1.

Illustration du cercle herméneutique. L'abduction permet d'extraire des observations des conclusions au niveau global, et les déductions à partir du global enrichissent le local.

Ces deux facettes de l'interprétation mettent en évidence la dualité du traitement du tout et de la partie dans l'interprétation : le mot et son entour pour le couple détermination/exclusion, l'auteur

comme individu ou au sein d'un groupe pour le couple divination/comparaison. Le cercle herméneutique (cf. figure 2.1) figure une démarche d'interprétation qui alterne le traitement de la partie et du tout. Le tout ne peut être compris sans en comprendre les parties et les parties ne peuvent être comprises sans le tout.

Pour Droysen [86] qui s'intéresse au processus d'interprétation de l'Histoire, l'interprétation est le résultat de plusieurs cercles qui ont cours en même temps pour négocier des espaces liminaux spécifiques à réduire. Ces cercles sont tous intriqués (*interlinked*). Bien qu'attaquant en parallèle des espaces différents, ils s'influencent mutuellement pendant l'interprétation. Ils peuvent également s'organiser de manière hiérarchique, *e.g.* Droysen compare la construction d'une interprétation à la construction d'un état : États fédéraux, les confédérations d'États, le système des États et le système mondial des États.

L'élucidation du sens passe alors par un *parcours interprétatif* [175], chemin cognitif réalisé par le sujet. Celui-ci se situe à un moment donné dans un état cognitif caractérisé par la position avec laquelle il regarde l'objet (niveau plutôt global ou local) et le questionnement qui se pose à lui comme direction du chemin à suivre. Et ce questionnement est résolu à l'aide des connaissances propres du sujet et de la compréhension de l'objet qu'il construit qui ne sont encore que partielles. La compréhension en cours de l'objet est le résultat de chemin parcouru jusque là s'accompagnant d'opérations de contrôle et de corrections progressives, d'un jeu interne de questions/réponses.

En résumé, le cercle herméneutique formalisé par Schleiermacher propose une grille explorant la syntaxe de l'objet et son auteur selon des aspects locaux et globaux. Le processus d'élucidation du sens suivi par le sujet interprétant est un parcours faisant circulairement s'intéresser le sujet aux aspects locaux ou globaux de l'objet. Dans cette approche de l'interprétation, Schleiermacher met de côté l'importance du troisième acteur qu'est l'interpréteur de l'objet (le sujet). En particulier, il apparaît que la méthode permet à un interpréteur d'extraire de manière déterministe le sens de l'objet. Or, l'interprétation n'est pas une explication de l'objet mais plutôt une *appropriation* [17]. La compréhension est atteinte lorsque qu'il y a effectivement appropriation de l'objet par l'interpréteur.

A.1.2 Appropriation et contextualisation de l'interprétation

Tout d'abord mettons en évidence que l'interpréteur est actif puisque d'une part il construit son chemin interprétatif et d'autre part celui-ci recrée l'œuvre de manière interne (dans son *registre* pour reprendre le terme de Schleiermacher) impliquant en cela ses propres connaissances antérieures. L'appropriation qu'il effectue conduit nécessairement à une relativité de l'interprétation de l'objet. Dans un second temps, nous sommes alors confrontés au constat que deux interprétations réalisées par deux personnes différentes se recoupent généralement en grande partie. La contextualisation apparaît comme une force imposée par un contexte (*e.g.* culturel) commun aux interpréteurs qui s'impose alors sur le résultat de l'interprétation.

La démarche interprétative progressive soulevée dans l'idée de parcours fait passer le sujet d'un état à l'autre au fur et à mesure de la résolution du sens de l'objet. Ce chemin est propre à l'interprétant qui ne renaît pas à chaque interprétation mais qui s'accompagne de toutes les interprétations qu'il a pu opérer auparavant en relation avec celle qu'il va réaliser.

Le concept d'intertextualité² [116] met bien en évidence que la compréhension d'une œuvre n'est pas indépendante des autres textes qui sont à la connaissance d'un lecteur. On peut résumer cela en disant que personne ne lit jamais le même texte. Hongre [116] prend l'exemple de la lecture de *Phèdre*, nous ne "recevons" pas ce texte de la même manière aujourd'hui que les lecteurs contemporains de Racine. C'est d'ailleurs pour cela que les acteurs doivent *interpréter* pour transmettre le sens original de la pièce de théâtre pour les publics d'aujourd'hui. Autre exemple, un texte parlant de "spleen" ne prendra pas le même sens pour le lecteur selon qu'il a lu Baudelaire ou non.

Le sujet s'investit ainsi activement dans la construction de la signification en reconstruisant intérieurement l'objet interprété. Pour Gadamer [98], interpréter une œuvre d'art, c'est faire une expérience et savoir recevoir. L'art est une expérience dans laquelle les signes et les figures sont reçus par l'interpréteur qui doit la **recréer** pour en atteindre une compréhension.

Par conséquent, une interprétation est subjective et difficilement partageable. On comprend bien que

²La notion d'intertextualité apparaît aussi bien dans l'acte d'écriture que dans celui de la lecture. On se concentre ici uniquement sur le second.

dans la mesure où l'expérience de chacun est individuelle, l'interprétation qui dépend de cette expérience est elle-même subjective. Dans l'absolu, pour partager une interprétation il faut au mieux la faire revivre à celui avec lequel on souhaite la partager, en sachant que même en la revivant, il ne la recevra pas nécessairement de la même manière.

Mais si l'interprétation est dépendante du chemin interprétatif suivi et de l'expérience propre du sujet, se pose alors la question de la maîtrise du sens véhiculé par l'objet. Il est certain que parfois les messages ne sont pas reçus de la façon dont il a été imaginé qu'ils le seraient. Mais en allant plus loin on peut se poser la question de la construction d'un sens commun ? Car malgré tout, nos interprétations quotidiennes se recouvrent étonnamment pour nous permettre en particulier de partager une compréhension commune du monde. Selon Bachimont [15], de même qu'il n'y a pas *unicité du sens*, il n'y a pas d'*infinité des sens*. Ce qui permet la construction d'un sens commun, c'est le partage d'une culture commune qui est un contexte à l'interprétation. L'appropriation co-implique un mouvement parallèle de contextualisation sociale.

Nous appelons **contextualisation sociale** l'implication du contexte culturel et ontogénétique dans lequel est un interpréteur lorsqu'il aborde son interprétation. C'est-à-dire que l'interpréteur ne peut être considéré comme une unité, mais que son interprétation est réalisée (*i.e.* influencée et perçue) par les autres éléments d'une société à laquelle il appartient et hérite des interprétations passées. On retrouve ici quelque chose de semblable au couple dual divination/comparaison de Schleiermacher. Le contexte social participe à la sélection des chemins interprétatifs de l'interpréteur de sorte à ce qu'ils s'inscrivent dans une logique sociale permettant la communication entre les individus. Pour E. Morin, "les liens entre individus d'une société animale, comme ceux des cellules d'un organisme, ne sont pas seulement de communication selon un code commun, ils comportent aussi une dimension communautaire, c'est-à-dire d'inclusion des êtres dans un circuit trans-subjectif"³.

Le contexte culturel participe à influencer l'interprétation. On peut penser que la culture est contenue dans l'expérience de l'interpréteur mais elle a la particularité d'être une expérience qui est partagée avec un grand nombre et que l'interpréteur participe également à construire. Par exemple, la morale est un élément culturel fort. Elle nous permet d'interpréter les actes de nos semblables. Et les personnes culturellement attachées à la morale judéo-chrétienne, n'interpréteront pas nécessairement les actes d'une personne de la même manière que celles attachées à une morale protestante.

Le contexte ontogénétique, dans le sens de l'héritage de certains facteurs biologiques sur des générations successives, participe également à une contextualisation de l'interprétation *du monde*. Les interprétations de l'Homme en tant qu'espèce vivante interprétant le monde héritent de son évolution. Celles-ci sont différentes de l'*interprétation* du monde réalisée par des espèces vivantes différentes.

A.1.3 Objectif inatteignable de cohérence

Nous pensons que le processus d'interprétation est un processus sans fin. De part la structure circulaire des cercles herméneutiques, le parcours interprétatif est lui-même circulaire, c'est-à-dire que l'interprétant passe et repasse sur un même objet de questionnement. Mais à chaque passage, son expérience passée lui donne un nouveau point de vue sur l'objet, affinant un peu plus à chaque étape l'interprétation.

On peut de toute façon se poser la question de savoir quand est-ce qu'on peut considérer qu'une interprétation est finie ? Dans la mesure où l'interprétation est une tâche qui consiste à réduire un espace liminal (ou plusieurs selon Droysen), alors l'interprétation est finie lorsque qu'il n'y plus d'espace : Tant qu'il existe un espace il reste de la place pour l'interprétation. Scheiermarcher écrit "clarifying what is vague is never ending".

Schleiermacher définit l'herméneutique comme une tâche devant éliminer toute trace d'incompréhension ("We could formulate the task of hermeneutics in negative terms : to avoid misunderstanding at every point" [183]). L'incompréhension vient, selon nous, d'une **incohérence** entre les connaissances préalables et l'interprétation de l'objet. La cohérence est définie comme une absence de tension. Il s'agit d'une absence de tension d'une part entre les éléments internes d'un ensemble, et d'autre part entre les éléments intérieurs et les éléments extérieurs de l'ensemble. Dans le premier cas, Il s'agit de la cohérence d'un ensemble d'éléments pris isolément. Cet ensemble est cohérent si il ne peut pas être produit de contra-

³Edgard Morin, La méthode. La vie de la vie.

dictions entre les éléments qui le composent. Dans cette définition, les relations entre les composants sont alors définis *a priori*. Dans le second cas, il s'agit d'une cohérence entre une représentation interne (réduite) d'un monde extérieur qui cherche à faire correspondre les extensions avec les intensions. Dans le cadre de l'opérationnalisation de l'interprétation de séries temporelles pour un système informatisé cette notion de cohérence pourra être explicitée.

Ce qui pousse un interpréteur à poursuivre son interprétation c'est d'une part une incohérence entre l'interprétation en cours de l'objet et ses propres connaissances (appropriation non satisfaisante), et d'autre part une incohérence entre l'interprétation en cours et l'objet (non adéquation de l'interprétation avec les observations qui en sont faites). Lorsque l'ensemble de ces trois entités (objet, interprétation et connaissances) ne sont pas en contradiction pour l'interpréteur, *i.e.* sont cohérentes, alors il n'a plus de raison de poursuivre son interprétation.

Ce qui peut arriver, c'est une modification de l'une des trois entités qui va introduire une perturbation, une tension, dans cet ensemble cohérent et qui va alors conduire à une nouvelle interprétation.

L'interprétation cherche donc à atteindre une cohérence qu'on sait être dans le cas général inatteignable. Construire une interprétation en effet, c'est construire quelque chose de fini et défini pour quelque chose d'infini et indéfini ("Construct something finite and definite from something infinite and indefinite", W. Iser [122] p.100). L'interprétation sera toujours incomplète face à l'objet à interpréter. Et donc, il existera toujours des tensions qui se présenteront pour remettre en question ce qui a déjà été construit comme interprétation.

En résumé, l'interprétation est un art d'élucidation du sens d'un objet qui implique un interpréteur dans un parcours interprétatif sans fin par lequel il est amené à créer et recréer l'objet dans un registre (déjà formé par ses interprétations antérieures) qui lui est propre.

Dans le cadre de ce travail, nous cherchons un moyen d'opérationnaliser un parcours interprétatif de sorte à l'aborder à l'aide d'outils informatiques. Nous identifions dans les caractéristiques d'une interprétation les mécanismes d'un phénomène émergent. C'est-à-dire que l'élucidation du sens d'un objet serait semblable à un phénomène d'émergence d'un "sens" de celui-ci.

Nous nous intéressons donc dans la section suivante à l'interprétation vue comme une émergence de "sens" qui nous donnera un point d'accroche pour opérationnaliser cette tâche.

A.2 Interprétation et émergence de "sens"

L'étude de l'interprétation dans le cadre dominant de l'herméneutique ne laisse pas apparaître le problème important de la création du sens. Ce qui constitue les textes, les mots, sont porteurs de sens. Ce qui nous intéresse dans la construction d'une interprétation, c'est comment les éléments de l'objet peuvent acquérir une signification, c'est-à-dire comment le signifiant et le signifié peuvent émerger de l'objet.

Dans cette section nous nous attachons à voir l'interprétation comme un processus qui permet de faire *émerger du sens* à partir des données. Ceci pose d'une part la question de la notion d'émergence qui nécessite d'être précisée et également la notion de *sens*. Nous nous appuyerons sur le travail de Petitot [169] pour proposer une caractérisation des moyens d'acquisition sémantique. Et finalement, nous présenterons des travaux qui s'intéressent à opérationnaliser (informatiquement) la construction du "sens" avec des approches semblables.

A.2.1 Émergence

Donner une définition de l'émergence n'est pas une chose aisée. Dans une première approche, un phénomène émergent est identifié lorsque le tout est plus que la somme des parties. Cette définition met en évidence l'idée de partie et de tout qui sont des "niveaux" différents du système. On parle également de niveaux *micro* et *macro* ou *local* et *global*. Mais on trouve dans la littérature une grande disparité de définitions sous cette même appellation [23, 74, 124, 149] qui viennent préciser ces idées.

Nous retenons comme définition de l'émergence : l'apparition d'un phénomène ou d'une entité engendrée par les interactions des entités à un niveau inférieur qui constituent ce système. L'entité de niveau supérieur présente la propriété de rétroagir sur ce système l'ayant engendré (on parle alors d'*immersion*).

- Les éléments importants qui caractérisent un phénomène émergent dans la définition retenue sont :
- L'existence de niveaux dans le système,
 - La dynamique du système qui trouve d'une part son origine dans les interactions entre entités et d'autre part qui est régulée par la rétroaction du phénomène émergent sur ce qui l'a produit,
 - L'existence d'un observateur capable de mettre en évidence une apparition. L'observateur peut être extérieur ou non au système.

Nous sommes conscients qu'en fixant une acception pour la notion d'émergence, nous opérons une sorte de réduction de l'ensemble des phénomènes qui prétendent à ce titre. La définition choisie pour l'émergence sert notre objectif de mieux comprendre l'interprétation, nous ne prétendons pas ici proposer une définition systémique de l'émergence.

Dans la suite de cette section, nous revenons plus en détails sur les trois caractéristiques de l'émergence.

Niveaux et hiérarchies En introduisant la notion de niveau dans un système, on se trouve de fait rattaché aux théories des systèmes hiérarchiques. Une organisation hiérarchique d'un système signifie qu'étant donné un niveau de résolution, le système est composé d'entités en interaction (entités de niveau inférieur) et est lui-même un élément d'un système plus large (entités de niveau supérieur) [164]. Ratzé et *al.* [176] présentent les concepts clés de la notion de hiérarchie rencontrée dans la modélisation de systèmes écologiques :

- Le concept d'**échelle** (spatio-temporelle dans le cas des systèmes écologiques). L'échelle peut être 1) un filtre de la perception d'un système, 2) une notion intrinsèque au système observé ou 3) une limite à la représentation du système.
- Le concept de **niveau** d'organisation est défini par l'identification de sauts dans le continuum des échelles. Ces sauts correspondent à des frontières entre les niveaux.
- Le concept de **holon** qui distingue la relation hiérarchique de la relation holonique : La notion d'holon [131] est définie comme une entité qui est également un composé d'éléments plus petits. L'idée illustrée par la figure 2.2 est qu'à l'intérieur d'un niveau, des entités ont des interactions privilégiées entre elles de sorte à former un tout avec une cohésion suffisante pour être abstraite comme une entité globale qui sera perçue comme unité à un niveau supérieur.
- Le concept de **contrainte** introduit l'idée que le comportement des entités d'un niveau N est limité à un ensemble de possibilités qui est contraint par le niveau $N + 1$. L'une des conséquences importante pour la suite est que la dynamique d'un système n'est pas simplement dépendante des interactions au niveau inférieur, elle l'est également des contraintes imposées par le niveau supérieur.

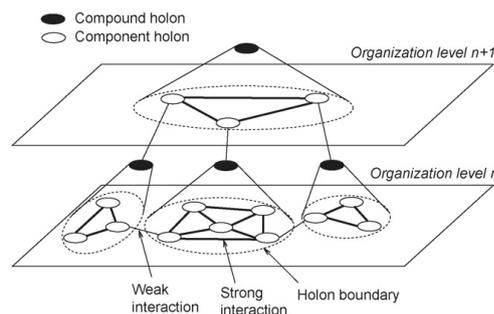


FIG. 2.2. Organisation hiérarchie en holon (Figure extraite de Ratzé et *al.* [176])

Dynamique du système La dynamique d'un système hiérarchique comporte donc à la fois une dynamique d'interactions entre les éléments d'un même niveau et une dynamique d'interactions entre les niveaux hiérarchiques.

La dynamique intra-niveau est de nature génératrice de formes tandis que la dynamique inter-niveaux est de nature à confiner la profusion de la génération. Un équilibre des deux est nécessaire pour observer un phénomène émergent. Nous prenons l'exemple des automates cellulaires pour illustrer la nécessité de

cet équilibre. L'étude des automates cellulaires⁴ permet de mettre en évidence qu'un phénomène émergent ne peut être identifié que dans des conditions particulières qui permettent un fonctionnement du système entre un régime chaotique et une régime périodique. En fonction des règles des automates cellulaires, le comportement global observé est associé à l'une des quatre classes suivantes :

- Classe I : Au bout d'un certain nombre d'itération, le système se trouve dans un point fixe, *i.e.* qu'on n'observe plus aucune modification de la grille de cellules.
- Classe II : Après une phase transitoire, l'automate se place dans un régime périodique. Les automates de classe I peuvent être vus comme des automates de classe II avec une période de 1.
- Classe III : Ces automates ont un comportement chaotique. Malgré leur caractère désordonné et leur instabilité locale, il existe une certaine forme de stabilité dynamique au niveau global (*e.g.* conservation de propriétés statistiques).
- Classe IV : Ces automates sont dans un état transitoire de longue durée et présentent une grande diversité de structures. L'exemple le plus célèbre des automates de classe IV est le "Jeu de la vie" défini par Conway.

Les automates de classes IV se situent entre les automates de classe II et les automates de classe III : Ils sont à la frontière entre l'ordre (le périodique) et le chaos. Selon nous, les systèmes de classe III ne sont pas suffisamment confinés et la profusion de la génération empêche une stabilité nécessaire à l'identification de formes émergentes. À l'opposé les systèmes de classe I et II ne disposent pas de suffisamment de dynamique locale pour que surviennent des formes complexes que l'on trouve dans les systèmes de classe IV et qui sont associées à de l'émergence.

La dynamique des éléments d'un même niveau est de nature auto-organisatrice. C'est à partir des interactions entre les éléments d'un même niveau que sont construites les entités holoniques par un phénomène d'*auto-organisation*. On parle d'auto-organisation lorsque la dynamique de l'ensemble des entités en interaction les conduits à s'organiser. Oudeyer [165] présente une large revue de structures naturelles dans lesquelles un phénomène d'auto-organisation peut être identifié : cellules de Bernard en mécanique des fluides, formation des dunes, structures cristallines ...

L'organisation d'un système sera plus importante lorsqu'il existera des structures de *cohésion* forte associant certaines entités du système. L'auto-organisation d'un système conduit à former ces structures holoniques qui apparaissent (ou *surviennent*) au niveau d'abstraction supérieur. Par conséquent, cette dynamique permet également des interactions entre les niveaux par ce qu'on a appelé une fonction d'émergence : apparition d'une entité au niveau supérieur par les interactions au niveau inférieur.

La dynamique inter-niveaux régule l'auto-organisation. La capacité d'un système auto-organisé à faire émerger des entités à un niveau supérieur ne répond pas entièrement à la définition forte de l'émergence retenue. Nous pensons que la dynamique d'émergence doit inclure également une fonction d'immersion qui, selon la théorie des systèmes hiérarchiques, va contraindre les états possibles atteignables par auto-organisation et ainsi contribuer à la dynamique globale du système. L'immersion agit donc sur l'auto-organisation en la facilitant et la dirigeant, *i.e.* en favorisant certaines solutions cohérentes par rapport à d'autres. Non seulement le phénomène émergent permet de faire survenir des structures à des niveaux supérieurs, mais cette survenance est en partie dirigée par les contraintes du niveau supérieur.

Dans la nature, différentes sortes de fonction d'immersion peuvent être identifiées :

- les facteurs environnementaux : par exemple les lois physiques globales telles que la gravitation qui influent sur la croissance des plantes et leurs structures.
- le contexte social : Steels et Kaplan [204] proposent un système dans lequel la construction d'un langage par des agents autonomes est contrainte par la nécessité sociale de communication entre les agents.
- la sélection naturelle : Oudeyer [165] lie le phénomène d'auto-organisation à la sélection naturelle. Dans ce travail, une partie de la structure de l'appareil phonatoire humain est dépendante de la nécessité de communication rencontrée par les humains.

Pour d'autres domaines dans lesquels des phénomènes émergents sont identifiés, on peut (quasiment⁵) toujours trouver un niveau supérieur qui englobe le système considéré et l'influence.

⁴Un automate cellulaire est composé d'une grille de cellules qui peuvent prendre différentes valeurs entières. On donne ensuite des règles de modification de la valeur d'une cellule en fonction de sa valeur et de celles de son entourage. L'application récursive de ces règles permet d'observer une dynamique globale de la grille de cellules.

⁵On met ici à part le système $\{Univers\}$.

L'influence d'un niveau supérieur sur le niveau inférieur (fonction d'immergence) se fait au moyen des contraintes de *cohésion* des entités en structure holonique. La cohésion est une propriété qualifiée par Collier *et al.* [66] comme la base d'une dynamique identitaire, *i.e.* une dynamique qui maintient certaines propriétés d'une structure dans le temps. Cette cohésion est manquante pour les automates cellulaires d'ordre III puisque les formes générées ne sont que très transitoires.

Collier [65] définit la cohésion comme la clôture des relations causales d'une partie des entités du système. Avec cette définition, la cohésion est une relation d'équivalence qui permet de partitionner les entités d'un niveau en structure holonique. Par conséquent, la fonction d'immergence participe à définir ce que sont les contraintes de cohésion d'une structure holonique et ainsi, elle influence les structures elles-mêmes.

Pour conclure sur notre point de vue sur la dynamique d'un système dans lequel on peut observer un phénomène émergent, on prend une métaphore énergétique. Une structure émergente est créée et maintenue d'une part par l'énergie dépensée au niveau inférieur pour organiser les entités (l'auto-organisation est un phénomène dissipatif d'organisation allant en sens inverse de l'augmentation de l'entropie) et d'autre part grâce à la régulation de la dépense d'énergie par l'introduction de contraintes du niveau supérieur (mise en cohésion des entités).

Observateur Dans la notion d'échelle, il est distingué les échelles intrinsèques et les dimensions relatives à un observateur (liées à son observation et à sa représentation du système). L'émergence peut donc prendre un caractère objectif ou subjectif.

Les échelles intrinsèques peuvent exister dans des systèmes où il est également question d'émergence de manière formelle. Ainsi d'après la théorie des catastrophes de R. Thom, un phénomène émergent est identifié par une bifurcation et l'émergence statistique est une propriété qui apparaît à partir d'une certaine masse critique d'entités interagissantes (*e.g.* la pression en mécanique des fluides). Dans les deux cas, il a été défini un critère objectif de l'apparition d'un phénomène émergent.

La part subjective du phénomène nécessite la présence d'un observateur. Et l'identification de l'émergence nécessite que celui-ci observe le système au niveau adéquat, *i.e.* celui dans lequel apparaît le phénomène. Prenons l'exemple bien connu d'une fourmillière, l'observateur ne pourra rien observer s'il considère les fourmis individuellement, il faut observer la fourmilière en entier. De même le caractère de nouveauté qui peut être attribué à un phénomène émergent dans certaines définitions nous apparaît comme subjectif pour l'observateur. Et donc pour la définition retenue la nouveauté n'est pas intrinsèque au phénomène émergent.

Les notions de niveau, de dynamique et d'observateur donnent un point de vue théorique sur l'émergence. L'opérationnalisation de ce phénomène par un modèle systémique va nous permettre par la suite de l'implémenter sur une machine.

Modèle systémique des triades Le modèle de triade est un modèle systémique introduit par Schwarz [185] et qui est également connu comme le modèle systémique de Neufchâtel⁶. Selon Schwarz, une triade (*cf.* figure 2.3) est constituée de deux entités en interactions réciproques et qui forment un tout émergent. Les deux entités interagissent et de leurs interactions émerge un tout, qui selon l'acceptation sera plus que les parties. La fonction d'émergence est alors représentée par les flèches sortant des entités vers le tout. La fonction d'immergence est représentée par les flèches inverses.

Ce modèle à l'avantage de représenter de manière simple les caractéristiques d'un phénomène émergent. Ce modèle de triade est hiérarchique :

- il est constitué d'entités en interactions qui sont simplement au nombre de deux (le fait de ne considérer que ce minimum d'entités permet de simplifier le modèle),
- il est composé de niveaux imbriqués : le tout peut à son tour être considéré comme une entité qui peut alors être en interaction triadique avec une autre entité du même niveau,
- il est holonique car les entités sont définies comme le composé de deux entités inférieures,
- il met en relation les niveaux par des fonctions d'émergence et d'immergence.

⁶Nous ne donnons ici que la partie élémentaire du modèle systémique de Neufchâtel, car à partir de ce système très simple, Schwarz décrit les étapes qui permettent de faire évoluer les deux entités à l'aide de simples interactions physiques vers un système qui construit sa propre unité.

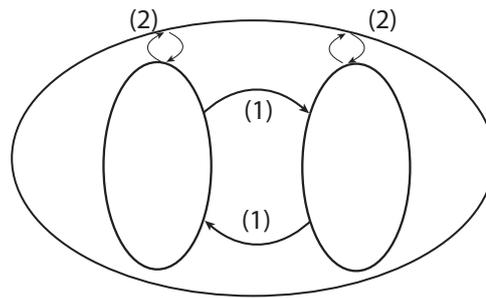


FIG. 2.3.

Triade de Schwarz [185] : Deux entités en interactions (liens (1)) qui forment un tout émergent. Les liens (2) figurent les fonctions d'émergence et l'immergence.

Parmi les organisations hiérarchiques de la littérature, l'organisation en triade est proche d'une organisation des systèmes multi-agents de type AGR (Agent, Groupe, Rôle) [93]. Une organisation en triade est une forme d'organisation hiérarchique d'agents. Le lecteur intéressé par plus de précisions pourra se reporter à la revue de Ratzé et *al.* [176] sur les hiérarchies dans les systèmes multi-agents.

On retrouve également des similarités avec le modèle de *hiérarchies enchevêtrées*, ou *boucle étrange*. Ce modèle explique l'apparition de phénomènes émergents par l'intrication entre deux niveaux. On trouve des exemples d'intrication entre deux niveaux en biologie. Par exemple, un gène permet d'exprimer une molécule (d'un niveau de complexité supérieur à l'ADN) mais d'un autre côté l'expression du gène peut être différencié par la présence d'autres molécules. Les molécules produites et les molécules influençant l'expression du gène peuvent également être liées dans un réseaux d'interactions complexes de sorte que le niveau des gènes et le niveau des molécules se trouvent intriqués. Et c'est de cette intrication qu'émerge en partie la complexité cellulaire. D. Hofstadter [115] illustre également ce phénomène par les gravures de M.C. Escher (*e.g.* Bildergalerie, 1956). Dans le modèle de triade, les entités en interaction sont à un même niveau, mais si à la place d'avoir des entités au même niveau, se sont des entités de niveaux différents, le modèle permet de représenter l'intrication des niveaux par une relation forte entre deux niveaux desquels émerge un phénomène à un troisième niveau.

De plus, la notion de triade dans laquelle seulement deux entités interagissent permet de représenter les situations de collaboration auxquelles nous allons nous intéresser (Homme/Homme ou Homme/Machine) ou la situation de l'observateur et d'un système. Ce sont également des situations dans lesquelles émergent des solutions à des problèmes insolubles par une partie seulement.

A.2.2 Émergence de "sens"

L'émergence de "sens" pose la question de l'émergence d'une signification pour un agent (un humain ou un système artificiel) à partir d'un objet. Les phénomènes précédemment présentés s'apparentent à des phénomènes d'émergence de *formes*. Mais la signification ne se réduit pas à la forme.

Constitution du "sens" La *forme* s'oppose alors au *sens* sur la base de la distinction entre le signifiant (le concept) et le signifié (la chose du monde). Pour nous, le "sens" d'un objet est la relation entre un contenu conceptuel (internalisation) et une chose du monde (externalisation). Ce qui caractérise l'information signifiante, c'est qu'elle renvoie toujours à autre chose qu'elle-même pour quelqu'un. Exprimé dans l'autre sens par Steels, "a meaning is a feature which is relevant in the interaction between the person and the thing being represented" [202]. Nous nous inspirons du structuralisme présenté par Petitot [169] pour identifier les éléments qui participent à la constitution du sens d'un objet. Nous dégageons ainsi quatre grands éléments qui participent à la construction d'une signification⁷ :

- La saillance;
- Le paradigmatique, *i.e.* la catégorisation et la différentiation;
- Le syntagmatique, *i.e.* la situation locale de l'objet;
- Le contextuel;

⁷Nous empruntons ici le vocabulaire de la linguistique.

La saillance est définie par Thom et Petitot [169] comme un ensemble de singularités perçues (ou anticipées en fonction d'actions). Les objets ou les événements (structures) sont saillants dans la mesure où ils émergent (au sens décrit dans la section précédente) comme des structures et s'individualisent.

Les formes saillantes sont des structures porteuses potentielles d'un sens. La notion de forme renvoie à la perception. Le sens ne peut en effet être attribué qu'à des objets qui peuvent être perçus et caractérisés. Et donc le sens d'un objet ne peut se constituer que sur ces éléments saillants. Les formes saillantes peuvent être des productions intentionnelles, *e.g.* les caractéristiques des signes d'un alphabet ou bien des formes issues de processus physiques ou biologiques émergents, *e.g.* les traces des fourmis. Dans le premier cas, ces formes ont un sens préexistant. Mais dans le second cas, les formes ne sont qu'un support auquel un interpréteur peut potentiellement associer un sens. Mais toute saillance qui émerge d'un processus physique n'a pas nécessairement du sens pour un observateur.

Thom et Petitot fondent leur théorie de la morphogénèse du sens sur la distinction entre saillance et prégnance. La prégnance est une valeur associée qui prend place dans une conception du monde (par un effet ontogénétique, individuel ou culturel). Les structures sont prégnantes dans la mesure où elles sont en résonance diffuse et infusive avec les autres. La prégnance est donc faite de relations entre les éléments saillants et peut être associée aux trois autres axes de constitution du sens (paradigmatique, syntagmatique, contextuel).

Le paradigmatique renvoie à la position d'un objet dans une catégorisation de l'ensemble des objets qui peuvent être rencontrés. Un objet appartient à une catégorie lorsqu'il a des ressemblances avec certains objets et des différenciations avec d'autres. L'appartenance à cette catégorie participe à sa caractérisation. Réciproquement, en étant associé à cette catégorie, l'objet en acquiert également du sens : d'une part ses caractéristiques particulières sont mises en avant par rapport aux autres, et d'autre part, il est associé aux situations du monde des autres objets de la catégorie. De nouveaux liens sont ainsi créés entre l'objet et des externalisations : du "sens", tel que nous l'avons défini est donc constitué ainsi.

L'axe syntagmatique renvoie à la position d'un objet dans son entour local. Un objet n'apparaît jamais seul, mais toujours localement en relation avec d'autres objets. L'appréhension de l'objet tient compte de ces relations. Ainsi, pour J. Miermont [153]⁸, "le phonème ne prend sens que dans une unité sémantique, le monème. Le monème n'est compréhensible que dans un mot complet, qui lui-même ne prend sens que dans une phrase". La dimension syntagmatique participe à la constitution du sens en créant de nouvelles relations à des choses du monde (celles qui entourent l'objet) et également en mettant en évidence des schémas de relations usuels (de l'ordre du paradigmatique).

Finalement, le contexte rentre également en jeu pour le sens lui-même. Pour le distinguer du syntagmatique, il est pris ici comme le contexte de constitution de la forme saillante et celui de l'attribution du sens par quelqu'un. En fonction de cette description générale dans lequel apparaît l'objet, les éléments peuvent prendre différentes significations.

Mise en évidence de l'émergence de "sens" Nous pouvons maintenant reprendre les trois grands points de ce qui constitue notre approche de l'émergence : l'existence de niveaux, une dynamique de construction de structures cohérentes et la question de l'existence d'un observateur.

Nous avons mis en évidence la position d'un objet selon les trois axes, paradigmatique, syntagmatique et contextuel qui situent systématiquement un objet comme placé au cœur de relations de différentes natures avec d'autres objets. Ces relations constituent des structures *émergentes* également de natures différentes (catégories, syntagmes ou contextes). Et par l'émergence de ces structures, émergent des relations entre les objets du monde.

La dynamique de la construction de ces structures et la question de la place d'un observateur renvoient aux processus d'interprétation qui ont été vus dans la section A.1 qui mettaient en évidence comment un sujet (observateur) réalisait un parcours interprétatif construisant en parallèle la cohérence entre les différentes dimensions de constitution du sens des objets à interpréter.

⁸Miermont [153] évoque dans cette citation la notion de contexte au sens large. Nous distinguons pour notre part ce qui tient du syntagmatique et ce qui tient du contextuel.

A.2.3 Exemples d'opérationnalisations de l'émergence de sens

L'identification de l'interprétation comme un phénomène d'émergence de sens permet d'envisager une opérationnalisation de l'interprétation en opérationnalisant l'émergence de sens.

Cette opérationnalisation sur la base des opérationnalisations des phénomènes émergents (de formes) permettrait d'attribuer à un système artificiel la capacité de *créer* du "sens". Nous nous intéressons donc ici à des travaux qui se sont intéressés à rendre des systèmes informatiques créateurs de "sens".

Pour Alain Cardon qui s'attache à construire une conscience artificielle, l'émergence de sens est ramenée à une émergence de forme. Cardon part du principe que générer des faits de conscience, c'est produire de la signification pour soi-même. Par conséquent, son objectif de construire une conscience artificielle est dirigé par la problématique de faire construire du "sens" à un système artificiel. Le système se compose d'une part d'un robot avec ses capteurs et ses effecteurs dans le monde physique et un cerveau artificiel. Ce dernier doit permettre de faire émerger du sens à partir du flux continu d'information produit par les capteurs⁹. Le cerveau artificiel est un système à plusieurs niveaux : le *substrat*, les *formes*, l'*espace de génération de sens*, et enfin le *sens généré*. Le cerveau artificiel [48] réifie les informations données sous la forme d'*agents objets* qui sont disposés dans un espace topologique qui constitue le *substrat*. La disposition des agents (leur présence et leur position dans l'espace) constitue une représentation de l'information. Les agents forment un réseau dynamique d'éléments computationnels élémentaires à la combinatoire élevée. Des *formes* sont alors construites par cette dynamique émergente. Ces formes sont des descripteurs de la morphologie et de la dynamique des agents dans le *substrat*. Chaque forme exprime une transformation réorganisationnelle d'une partie du substrat. Elles sont représentantes de caractères du substrat et qualifiées pour cela de *signifiantes* (localement). L'*espace de génération du sens* est l'espace des formes possibles dans lequel on peut localiser les *formes* qui ont été générées par le processus sous-jacent. Ces *formes* sont alors considérées comme du *sens généré*. Par conséquent, la notion d'émergence de sens est ramenée à l'apparition d'une forme organisationnelle (morphologique et dynamique) localisée dans un espace topologique de possibles, l'*espace de génération du sens*.

S. Lahlou [134] propose une formalisation de l'interprétation de classes de mots faisant sens dans un corpus textuel. L'accent est mis dans cette heuristique sur la dimension paradigmatique de la constitution du sens. Pour l'auteur, l'interprétation des classes est "un processus par lequel l'analyste attribue un *sens* aux classes obtenues à partir de l'analyse des traits lexicaux typiques des unités regroupés dans chaque classes". De l'analyse de cette pratique, il tire une démarche heuristique générale d'interprétation composée d'étapes d'observations subjectives des données, d'extractions de traits discriminants pour la construction de classes homogènes signifiantes et de vérifications de l'adéquation de ces modèles avec les données. L'homogénéité d'une classe est ici une concrétisation de la notion de cohérence d'une structure catégorielle.

Steels et al. [201] s'intéressent à la construction d'un langage par une communauté d'agents informatisés, les *Talking Heads*¹⁰. La création de ce langage s'apparente à une création de sens ancré par la perception [199]. Chaque *Tête parlante* est un agent informatique doté d'une caméra numérique¹¹. La caméra lui permet de voir un tableau blanc sur lequel des formes sont disposées. Ces formes peuvent être caractérisées par plusieurs caractéristiques saillantes pour les *têtes* :

- la couleur, le niveau de gris moyen,
- la forme géométrique (carrée, rond, ...),
- la position

Chaque agent peut explorer les formes à sa disposition en commandant la caméra mobile. Par un jeu d'interactions, sur lequel nous reviendrons plus en détails dans la section C.2.4 de ce chapitre, les agents construisent un langage commun fait de concepts.

Pour construire les éléments du langage, les *têtes* disposent d'une capacité de catégorisation à partir des caractéristiques saillantes. À partir de cette catégorisation, des "mots" sont générés. Par exemple [204], lorsque l'agent veut parler d'un carré rouge du tableau, il ne génère bien évidemment pas le

⁹Cardon pointe le fait que l'incarnation physique n'est pas obligatoire pour construire ce qu'il nomme une conscience artificielle, mais qu'il est indispensable que le cerveau artificiel soit alimenté en continu par un flux d'information.

¹⁰Plus d'informations sur les *Talking Heads* : <http://talking-heads.csl.sony.fr/>

¹¹En fait, chaque agent peut se "téléporter" dans l'une des nombreuses caméras numériques qui sont en réseaux dans différentes localités du globe. Mais à un instant donné, une *tête* n'utilise qu'une seule caméra.

mot “carré rouge” mais “malewina” qui *signifie* [UPPER EXTREME LEFT LOW-REDNESS]. Par cette méthode paradigmatique, les *Talking Heads* construisent en même temps le signifiant et le signifié (par la catégorisation des formes qui est réalisée). La dimension contextuelle de la signification est également présente puisque le langage qui est généré est également soumis aux échanges avec d’autres *Talking Heads* qui initialement ont leur propre façon de *signifier* les choses et par leurs interactions, des *mots* consensuels sont construits permettant à la communauté d’agents d’échanger à propos du même concept. En prolongation de ce travail, Steels et *al.* se sont également intéressés à faire générer une grammaire à partir des concepts partagés [200]. L’utilisation d’une grammaire permettrait d’ajouter une dimension syntagmatique à la signification des concepts. Cette extension pose encore des difficultés.

En résumé, l’interprétation est un processus d’élucidation du sens d’un objet. Le sens est acquis à partir des éléments saillants d’un objet selon les dimensions paradigmatiques, syntagmatiques et contextuelles. L’interprétation est un processus qui va permettre de faire émerger le sens par la mise en cohérence de ces différentes dimensions. L’interprétation est un acte dont le sujet est moteur. Elle implique un interpréteur dans un parcours interprétatif qui le conduit à recréer le sens pour lui-même, ce sens lui est alors propre et non directement partageable. Le parcours interprétatif est une exploration de l’objet guidée par le sujet. Ce biais agit comme une fonction d’immersion sur le processus d’émergence de sens et c’est par ce biais que le sens construit est individualisé.

B Interprétation des séries temporelles

B.1 Éléments d’une interprétation de séries temporelles

B.1.1 Description des *atomes* d’une interprétation de séries temporelles

L’interprétation de séries temporelles multivariées, traces du comportement physiologique d’un patient, permet au clinicien d’acquérir une compréhension *a posteriori* des situations cliniques qui ont été observées. Les notions d’évènements et de scénarios introduites dans le chapitre 1 constituent les éléments sémantiques élémentaires qui construisent l’interprétation de séries temporelles.

La compréhension de l’enregistrement d’un patient doit permettre aux cliniciens de savoir :

- Ce qu’il s’est passé, *i.e.* quels sont les évènements qui ont été observés? L’interprétation peut se contenter de noter que “quelque chose” de remarquable c’est passé sans pour autant identifier la nature exacte des évènements.
- À quel moment ces évènements se sont passés? L’observation d’un évènement doit donc être rattachée à un instant ou à une période de l’enregistrement.
- Pourquoi ces évènements se sont-ils passés? Les liens “causaux” ou “temporels” entre les observations doivent être mis en évidence. Un *lien temporel* entre deux évènements indique au clinicien qu’un évènement E_2 s’est déroulé, par exemple, 3s. après un évènement E_1 . La nature causale d’un lien est une relation plus forte qui indique que si E_2 s’est produit, c’est à cause de l’occurrence de E_1 (avec ou sans contraintes temporelles).

Prenant en compte ces besoins pour construire une interprétation nous définissons les différents éléments atomiques qui constituent l’interprétation d’une série temporelle multivariée, *i.e.* l’enregistrement d’un patient :

- Les **segments** : Un segment localise un intervalle de temps pendant lequel on observe sur une série temporelle une trace “remarquable” sans pour autant être capable de la qualifier.
- Les **symboles** : Un symbole localise un intervalle de temps correspondant à l’occurrence d’un évènement sur une série temporelle. L’évènement est un élément de référence consensuel pour les cliniciens. Il fait référence aux connaissances propres du clinicien sur les conditions d’apparition et les implications techniques, physiologiques ou médicales.
- Les **scènes** : Une scène associe plusieurs évènements d’un enregistrement avec un scénario. Le scénario est, comme les évènements, un élément de référence consensuel pour les cliniciens. Une scène fait intervenir plusieurs évènements liés par des relations temporelles ou causales. Par conséquent, les scènes permettent de contenir l’information causale entre l’occurrence d’évènements (*i.e.* des symboles) de l’interprétation.

Nous mettons en évidence que les segments et les symboles sont associés à des séries temporelles

univariées. En effet, même si un évènement physiologique n'est pas nécessairement associé à un signal en particulier, dans des cas pratiques il peut être important de profiter de la précision de l'association d'un évènement à une voie. Cela nécessite alors d'identifier le même évènement par plusieurs segments ou symboles avec les mêmes instants et durées dans plusieurs séries d'un même enregistrement. Par exemple, la scène fréquence de débranchement d'un appareil de mesure sensible tel qu'un capteur de saturation d'oxygène est spécifique uniquement à cette voie et peut y être identifiée. C'est en permettant de préciser que le débranchement est associé uniquement à une série qu'on peut mettre en évidence la non influence de cet élément sur les autres voies.

Les segments et les symboles sont définis sur un support temporel non nul, *i.e.* sur des intervalles. À l'opposé, ils auraient pu être instantanés. Le choix entre logique temporelle descriptive instantanée ou une logique à base d'intervalles est ici sans conséquence sur la sémantique qui peut être portée par les atomes de l'interprétation. La supposition qui est faite est que quelque soit la nature de l'évènement, il a une durée dans le temps pendant laquelle il se manifeste.

Finalement, l'interprétation est une activité contextuelle. Des références au contexte de l'enregistrement du patient sont référencées dans l'interprétation. Les éléments contextuels incluent d'une part des informations invariantes pendant l'enregistrement (*e.g.* les conditions climatiques, la pathologie du patient,) ou des informations dynamiques (*e.g.* la visite de la famille au patient).

B.1.2 Sens porté par les éléments de l'interprétation

En plus de répondre au besoin de l'interprétation d'un clinicien, ces éléments d'une interprétation sont également en relation avec les dimensions de la construction du sens qui ont été présentées dans la section A.2.2. Ils comportent donc selon nous un certain caractère générique. Ces éléments sont porteurs d'un sens qui est extrait des séries temporelles par le processus d'interprétation :

- Les segments permettent de repérer les informations saillantes dans les séries temporelles qui sont potentiellement porteuses d'un sens.
- Les symboles permettent d'établir des similarités ou dissimilarités entre les parties des séries. Ils intègrent donc la dimension paradigmatique du sens à l'interprétation
- Les scènes réalisant des liens entre les symboles localement situés dans l'objet à interpréter font porter du sens syntagmatique à la fois aux scènes et aux symboles qui constituent ces scènes.

Bien évidemment, les évènements et les scénarios qui sont décrits ici comme des connaissances expertes des cliniciens sont à rapprocher des modèles recherchés dans l'objectif de l'amélioration des systèmes de monitoring (*cf.* chapitre 1). Les symboles faisant référence à des évènements, un symbole porte un peu du sens de l'évènement auquel il est associé. De même, les scènes sont des instances de scénarios.

Par l'interprétation, l'évènement prend un sens concret en s'incarnant dans des situations réelles (les enregistrements faisant référence à des cas réels de patients), et les symboles et scénarios sont alors également porteur du sens clinique véhiculé par ces évènements. Ce sens (paradigmatique) est uniquement accessible à un clinicien. En effet, seul un clinicien pourra comprendre et enrichir le symbole ou la scène de cette connaissance. Son expérience clinique lui a fait s'approprier ce sens.

B.1.3 Abstraction des données

Les trois types d'éléments atomiques peuvent être utilisés pour construire des abstractions de l'enregistrement d'un patient (ou des séries qui le composent) :

1. Les séries segmentées : Les séries temporelles sont simplement enrichies des segments. Elles focalisent l'attention uniquement sur les parties segmentées, les autres parties ne sont pas significatives. Ces parties peuvent être écartées du reste de l'interprétation.
2. Les séries symboliques : Les séries temporelles sont abstraites comme une suite de symboles obtenus par la concaténation des symboles identifiés dans une série temporelle (univariée). Un enregistrement est alors une série symbolique multivariée.
3. Les enregistrements "scénarisés" : Les enregistrements sont abstraits comme des ensembles de scènes qui ont été identifiées.

Ces abstractions sont hiérarchiques puisque les enregistrements "scénarisés" sont plus abstraits que les séries symboliques dans la mesure où elles sont une représentation plus concise d'un enregistrement et

plus informative car porteuse du plus de sens. De même, les séries symboliques sont plus abstraites que les séries segmentées puis les séries temporelles brutes.

B.2 Interprétation de séries temporelles

B.2.1 Interprétabilité des traces de systèmes dynamiques

I. Weiss [222] insiste sur le fait qu'il n'y a pas d'interprétation s'il n'y a pas un sens caché à découvrir dans l'objet. On présuppose donc que les abstractions que nous cherchons peuvent être extraites des données dont nous disposons. Cette présupposition s'appuie sur deux hypothèses.

La première hypothèse est que des fonctionnements typiques régissent le fonctionnement d'un patient (ou d'un groupe de patients bien spécifié) et que ces événements sont identifiables à partir des données à disposition. Ceci revient à supposer d'une part l'existence de ce que l'on cherche et d'autre part l'existence de signes pouvant le révéler.

En s'appuyant sur les connaissances en physiologie et en médecine, il est raisonnable de penser que la première hypothèse est faible puisqu'elles permettent de rapprocher le fonctionnement de corps humain de celui d'une *machine*. Cette machine est certes complexe, mais son étude permet d'en déceler des fonctionnements systématiques (inter-patients et intra-patient). Dans un contexte médical où les patients sont variés, la systématisme des fonctionnements peut être remise en cause, la démarche d'interprétation pourrait alors intégrer un niveau supplémentaire d'identification et de sélection (démarches de nouveau duales) de groupes de patients *homogènes*.

La seconde hypothèse consiste à supposer que des traces de fonctionnements sont effectivement identifiables au travers des séries temporelles que nous traitons. Cette hypothèse est forte. Une interprétation est dépendante des données à disposition. Nous ne pouvons pas garantir que les séries temporelles permettent de distinguer les indices nécessaires pour réaliser une interprétation en correspondance avec les faits. Nous sommes tributaires d'une part de la qualité de l'acquisition, et d'autre part des capteurs eux-mêmes qui introduisent leur propre biais et dont la connaissance est nécessaire pour pouvoir en tenir compte dans l'interprétation. Par exemple, le signal de débit respiratoire d'un patient ventilé peut être influencé par la longueur du tube de ventilation utilisé.

Sous ces deux hypothèses, la tâche d'interprétation de signaux physiologiques de patients en unité de soins intensifs peut être envisagée.

B.2.2 Tâche d'interprétation de traces de systèmes dynamiques

L'interprétation de séries temporelles provenant d'enregistrements du fonctionnement d'un système dynamique diffère significativement des interprétations qui peuvent être faites des textes, discours, oeuvres d'arts ou de l'Histoire dans le sens où les signes sont produits sans intentions. On peut cependant remarquer que l'instrumentation qui permet d'obtenir ces signes est intentionnelle. En particulier, la simple présence de certains signaux spécifiques est significative.

Cependant, nous inscrivons cette tâche dans la lignée des interprétations qui ont été présentées dans la section A précédente. Il s'agit bien d'une interprétation dans le sens où on cherche à comprendre les informations représentées dans les enregistrements, *i.e.* à en élucider du sens. Au regard des caractéristiques de l'interprétation qui ont été dégagées de la section précédente, on peut alors dire que l'interprétation de séries temporelles :

- est un processus exploratoire et non-téléologique. La notion de parcours interprétatif s'applique aussi bien dans ce cas.
- est subjective. L'interprétation est dépendante de l'interpréteur qui la réalise.
- est un processus émergent permettant de faire construire et mettre en cohérence à la fois les éléments de l'interprétation et les définitions des événements et scénarios.

L'interprétation consiste à construire d'une part les modèles¹² d'événements et de scénarios, et d'autre part les abstractions des séries temporelles. La figure 2.4 illustre les mécanismes qui permettent sa réalisation.

Les éléments d'interprétation sont construits par niveaux de plus en plus abstraits :

¹²Pour un humain, un modèle est la représentation d'une catégorie, tandis que pour une machine, on parle de modèles *informatiques*, c'est-à-dire une représentation (réductionniste) des données ou connaissances sous forme informatique, calculable.

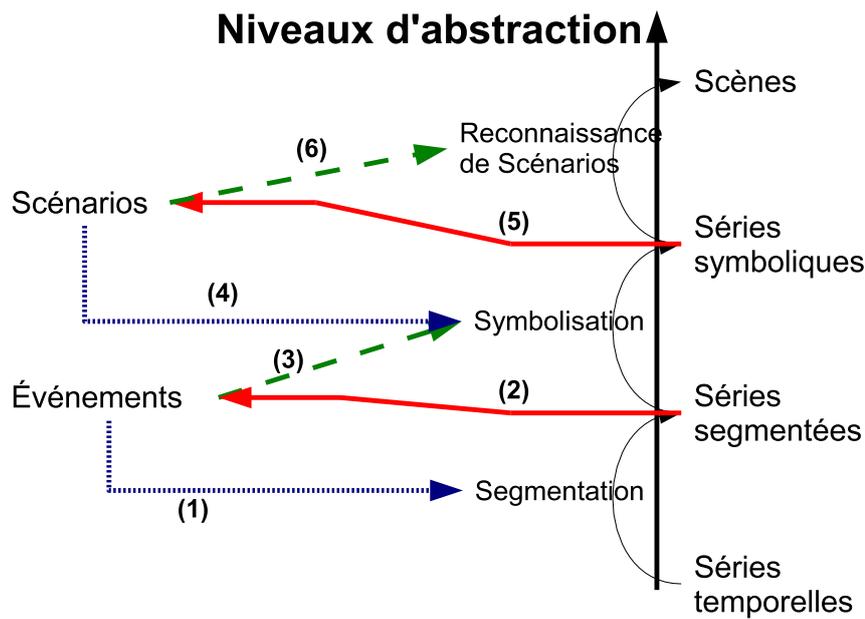


FIG. 2.4. Schéma général d'une interprétation de séries temporelles.

- la segmentation permet de construire les séries segmentées (par la construction de segments) à partir des séries temporelles,
- la symbolisation permet de construire les séries symboliques (par la construction de symboles) à partir des séries segmentées,
- la reconnaissance de scénarios permet de construire les scènes.

Ces processus utilisent les modèles d'évènements et les scénarios (flèches 3 et 5 en tirets sur la figure 2.4). Les modèles d'évènements servent (flèche 3) à la symbolisation des séries segmentées puisqu'un symbole permet d'associer des segments à des évènements. Les modèles de scénarios servent (flèche 5) à la reconnaissance de scénarios en associant des ensembles de symboles des séries symboliques avec des scénarios.

D'autre part, le système construit également les modèles à partir des données (flèches pleines 2 et 5 sur la figure 2.4). La flèche 2 montre que les modèles d'évènements sont construits à partir des segments d'intérêt identifiés dans les séries segmentées (par des méthodes de catégorisation des segments) et la flèche 5 montre que les scénarios sont identifiés à partir des séries symboliques exemples (par l'extraction de relations typiques entre les symboles).

Les modèles construits servent également à affiner les interprétations par un mécanisme de rétro-action entre les niveaux (flèches pointillées 1 et 4 sur la figure 2.4). Les modèles d'évènements sont utilisés (flèche 1) pour retrouver des segments dans les séries temporelles qui peuvent être intéressants pour affiner les modèles d'évènements. Les modèles de scénarios permettent de retrouver des symboles qui peuvent être intéressants pour affiner les modèles de scénarios.

Au travers des liens forts et multidirectionnels entre les modèles et les données, ou entre les niveaux d'abstraction, les éléments peuvent se mettre en cohérence pour constituer l'interprétation des séries temporelles.

B.2.3 Opérationnalisation de l'interprétation de séries temporelles

L'opérationnalisation de l'émergence au moyen du modèle de triade nous permet maintenant d'opérationnaliser l'interprétation de séries temporelles (comme phénomène émergent) à l'aide de ce modèle systémique.

L'organisation des traitements de la figure 2.4 fait apparaître deux étages de traitements, correspondants aux deux types de modèles, et ces étages présentent des similarités. Un premier niveau (dit *symbolique*) s'articule autour des modèles d'évènements et consiste d'une part à construire ces modèles

et d'autre part à traduire les séries temporelles brutes en séries symboliques. De manière indépendante, un second niveau se distingue en s'articulant autour des scénarios. Il consiste d'une part à construire ces scénarios à partir des séries symboliques (niveau hiérarchique supérieur au niveau **symbolique**) et d'autre part à identifier les scénarios dans les séries symboliques.

Chacun de ces deux niveaux se compose de deux tâches fortement couplées d'une part de construction de modèles (modèles d'évènements ou de scénarios) et d'autre part d'utilisation de ces modèles (symbolisation ou reconnaissance de scénarios) pour construire les abstractions. Dans ces deux tâches, on retrouve les deux tâches élémentaires du cercle herméneutique présenté dans la figure 2.1 avec un mouvement d'abduction (la construction de modèles) et un mouvement de déduction (la construction des abstractions). Une interprétation peut survenir du couplage fort entre ces deux tâches élémentaires. Le formalisme de triade (deux entités en interaction pour former un tout) introduit dans la section A.2.1 permet de figurer le couplage de ces deux tâches (*cf.* figure 2.5).

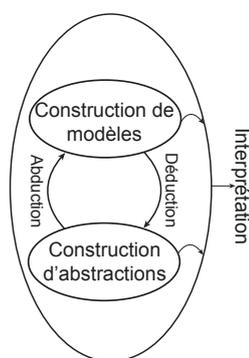


FIG. 2.5.
Triade : Brique élémentaire pour la construction d'une interprétation.

Nous utilisons donc le formalisme de triade comme brique de base pour avancer vers l'opérationnalisation de l'interprétation de séries temporelles. Ainsi, la figure 2.6 donne une vue conceptuelle globale de l'interprétation, organisée comme une hiérarchie de triades. On retrouve dans ces différentes triades les tâches à accomplir pour la réalisation d'une interprétation selon la figure 2.4. Globalement, l'interprétation est opérationnalisée par une triade constituée de deux entités de niveau inférieur (Traduction symbolique et Scènes) qui sont elles-mêmes des triades. La triade Traduction symbolique est constituée des triades de Segmentation et de Construction de modèles. La triade de Scènes est constituée des triades de Reconnaissance de scénarios et de Construction de scénarios.

Une triade permet de faire l'articulation entre la construction des modèles et leur utilisation dans l'abstraction des séries temporelles. Et c'est de la dynamique élémentaire des interactions réciproques entre triades qu'est organisée la dynamique globale émergence de l'interprétation :

- Le principe de cohérence entre la production des abstractions et la construction des modèles régule la dynamique élémentaire d'une triade : plus il y a de tensions entre les modèles et les annotations, plus les interactions seront nombreuses entre les deux triades pour permettre de les réduire.
- En créant des relations fortes entre niveaux, en incluant des rétroactions, une triade enchevêtre les niveaux de l'organisation hiérarchique naturelle des traitements.

Nous réalisons ici un glissement de la notion originale de Triade présentée par Schwarz. En particulier, nous y introduisons une dissymétrie entre les entités constituant une triade qui n'existe pas dans le modèle d'origine. Cette dissymétrie a pour origine des tâches de la hiérarchie d'abstraction qui sont à des niveaux différents, et en le couplant dans une triade, on enchevêtre les niveaux : Il n'y en a plus un niveau supérieur à l'autre. Par exemple, initialement la construction des modèles d'évènements succède à la segmentation, par le rebouclage du traitement dans la triade de Traduction symbolique, ces niveaux se replient sur eux-mêmes.

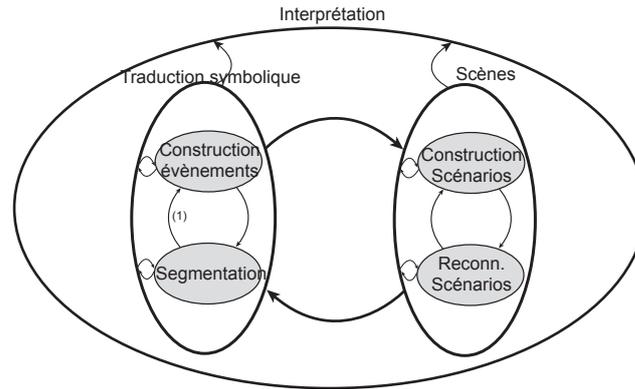


FIG. 2.6. Opérationnalisation de l'interprétation de séries temporelles sous la forme d'une hiérarchie de triades.

En conclusion de cette partie sur l'interprétation de séries temporelles, nous pouvons dire que la construction des modèles est donc une tâche intégrée à la réalisation d'une interprétation (par un utilisateur ou par une machine). Par conséquent, la réalisation d'une interprétation implique la tâche de construction des modèles que nous nous sommes donnés comme objectif dans le chapitre 1.

Ceci constitue donc l'originalité de notre approche de construction des modèles. Elle consiste à faire réaliser une interprétation par une machine ou par un clinicien. L'appropriation qu'ils se font des données interprétées va les conduire à construire leurs propres modèles des événements et des scénarios. Ainsi, la tâche d'interprétation permet à un utilisateur de mieux formaliser ses connaissances des données et permet à la machine de construire des modèles informatiques.

Pour l'interprétation de séries temporelles, nous avons proposé une opérationnalisation qui permet à un clinicien ou une machine de construire des modèles d'événements et de scénarios. Les modèles de la machine pourront servir à détecter les événements et les scénarios qu'ils formalisent dans un cadre de monitoring en reproduisant en ligne le processus d'abstraction des séries temporelles.

Cependant, dans l'interprétation vue comme un processus émergent, il apparaît toujours la question de la validité pratique du résultat de l'interprétation du système. Ce résultat peut être cohérent pour le système sans pour autant être intéressant en pratique. La collaboration avec un clinicien capable d'imposer une contrainte sur le processus émergent doit permettre de faire réaliser par le système une interprétation qui sera à la fois cohérente (donnant des modèles utilisables en pratique) et qui aura un sens clinique.

C Collaboration Homme/Machine

Selon Black et *al.* [28], une situation peut être qualifiée de collaborative lorsque deux agents ou plus travaillent ensemble dans un environnement commun pour améliorer la réalisation par l'union de leurs buts. Dans ce travail, nous nous intéressons plus particulièrement à la collaboration Homme/Machine.

C.1 Collaboration et Collaboration H/M

La collaboration et les moyens permettant de la faciliter ont beaucoup été étudiés mais principalement entre humains. En particulier, le récent domaine des CSCW (Computer Supported Cooperative Work) s'intéresse à la place de l'outil informatique pour faciliter la collaboration entre les humains (Voir [224] pour une introduction aux CSCW). Les études principalement sur la collaboration entre humains proposent un premier point de vue pour approcher la problématique de la collaboration homme/machine. Ainsi Clarke et Smyth [64] introduisent une approche de la collaboration basée sur les principes de la coopération entre humains proposée initialement par Marwell et Schmitt [145] :

- La collaboration est orientée vers la réalisation d'un but commun ;
- Les collaborateurs trouvent chacun un bénéfice (*reward*) à accomplir le but commun ;
- Les questions et les réponses soulevées doivent être connues de tous ;

- Des normes régissent la coopération entre les parties ;
- Une certaine coordination dans la réalisation de sous-tâches doit organiser la collaboration.

Mais ce point de vue sur la collaboration homme/machine est qualifiée par Salembier [180] comme une “illusion mimétique”. Deux critiques sont faites à l’approche mimétique de la collaboration homme/machine. D’une part celle qui veut faire concevoir la collaboration homme/machine en considérant systématiquement les humains comme un modèle pertinent à la conception d’une machine. Sauf pour des tâches bien définies, une telle approche semble utopiste. Ce paradigme “mimétique” a été en particulier utilisé dans les systèmes experts pour lesquels les concepteurs cherchaient à introduire toute la connaissance de l’expert de manière formelle pour que la machine puisse reproduire les raisonnements humains.

La dissymétrie évidente entre la machine et l’humain peut être alors vu comme un avantage pour la collaboration en présentant celle-ci comme le moyen de profiter de la complémentarité des compétences des humains et des machines. Classiquement, les machines se voient attribuer une compétence supérieure au traitement de masse et les humains des capacités à la création de nouvelles solutions.

C’est à ce niveau qu’intervient la seconde critique de Salembier. Elle remet en cause l’attribution rapide et définitive des compétences de l’humain et de celles de la machine (procédure du type MABA-MABA : “Men Are Better At-Machine Are Better At” [75]) lors de la conception d’un système en interaction avec un humain. C’est à partir de cette dernière critique qu’il paraît intéressant d’avoir une approche spécifique de la collaboration homme/machine par rapport à la collaboration entre humains.

Le problème de la collaboration homme/machine ne pouvant se rapporter entièrement aux collaborations entre humains, il faut donc l’aborder spécifiquement. Il semble alors naturel de s’intéresser aux thématiques des Interfaces Homme-Machine (IHM) qui traitent de la relation entre hommes et machines. En particulier, les IHM proposent des moyens pour assurer la transmission d’informations entre le système et la machine. Ce problème se pose d’autant plus qu’on ne peut pas imaginer de collaboration sans échange d’informations entre les parties. Mais une relation collaborative entre les machines et les humains ne pose pas uniquement des questions de l’interface elle-même [114]. Dans une situation collaborative, les machines ne sont plus considérées comme des outils, mais plus comme des agents capables d’une certaine autonomie et d’une forme de raisonnement.

La problématique qui est soulevée par la collaboration homme/machine est donc double. Tout d’abord, elle soulève le problème de la compréhension mutuelle, nécessaire à l’échange entre les agents, et ensuite le problème de la définition de caractéristiques de conception d’un système partenaire d’un humain, *i.e.* qui permette de construire une situation collaborative.

Le problème de la compréhension mutuelle entre les collaborateurs est une question essentielle pour arriver à une collaboration effective. La disparité “cognitive” d’une machine et d’un humain exacerbe cette problématique. La figure 2.7 illustre le problème d’une coopération sans compréhension mutuelle. L’agent de police cherche à aider le passant à retrouver son chemin, mais ce dernier ne possède pas une connaissance des lieux aussi précise que l’agent. La description qui lui est faite lui est donc incompréhensible.

Le second problème qui se pose est que si la machine n’est plus un outil : qu’elles doivent être ses caractéristiques pour assurer un bon “couplage” avec un humain ?

Dès lors, les propositions de conception d’une collaboration avec un humain considèrent la machine avec laquelle il faut collaborer comme une entité autonome (agent) ayant ses propres compétences. Les systèmes peuvent alors se caractériser par leurs approches pour résoudre les deux problèmes de la collaboration homme/machine suivants :

- Quelles doivent être les caractéristiques d’une machine à coupler à un utilisateur pour obtenir une situation collaborative ?
- Comment est résolu le problème de la compréhension mutuelle entre la machine et l’humain ?

Nous revenons maintenant sur ces questions à partir de quelques approches existantes puis nous donnons dans la section suivante les approches que nous avons explorées dans cette thèse.

C.1.1 Caractéristiques d’une machine pour une collaboration Homme/Machine

La construction d’une collaboration entre un homme et une machine suppose la construction d’une sorte de “couplage” ou de symbiose entre les deux agents. Si cette situation est connue entre humains, on s’interroge ici sur les caractéristiques qu’une machine doit posséder pour approcher ce type de situation

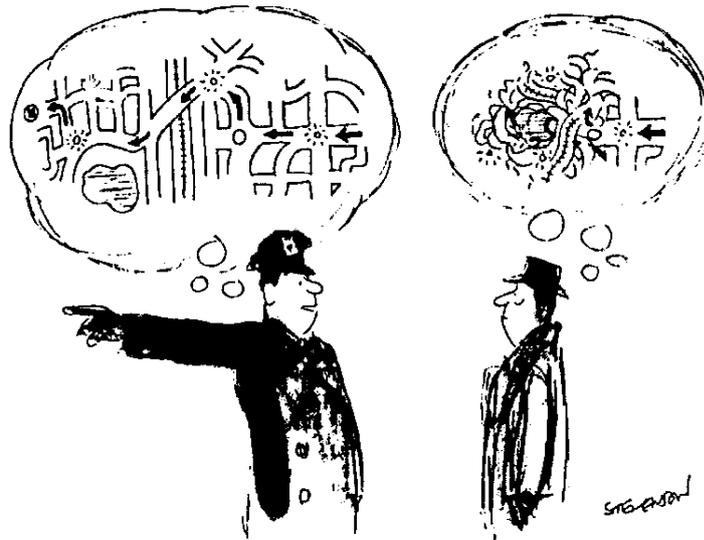


FIG. 2.7.

Illustration de l'échec d'une collaboration à cause d'un problème de compréhension. (Dessin par Stevenson, The New Yorker Magazine, 1976, extrait de [94])

entre elle et l'homme.

Systèmes cognitifs associés La notion de système cognitif associé (*joint cognitive systems*) a été introduit par Woods et *al.* [225] comme une alternative aux systèmes experts. Toute l'attention est portée sur l'aspect cognitif du couplage entre le système et l'humain. L'objectif d'un système conçu sur les principes de Woods est de suppléer les humains de sorte à réduire leur charge cognitive en situations dynamiques complexes.

Nous identifions quatre pré-requis forts des systèmes cognitifs qui dirigent la conception d'un tel système :

- Le partage de représentations cognitives ;
- Un consensus sur la notion contextuelle de résolution de problèmes ;
- Une aptitude à suivre le raisonnement de l'autre ;
- Une capacité à échanger des explications ;

Les deux premiers pré-requis sont de très haut niveau cognitif et il est difficile d'imaginer à ce jour les méthodes informatiques permettant de réaliser même partiellement ces tâches. Brézillon [37] propose une approche de génération contextuelle d'explications comme solution. Le contexte est alors défini en partie par le raisonnement suivi et perçu de l'utilisateur.

Systèmes à initiatives mixtes La notion de système à initiatives mixtes [68, 118] met en avant le fait que l'utilisateur comme le système peut être à l'origine d'une interaction. Dans les systèmes informatiques usuels, les utilisateurs sont seuls maîtres du déroulement de l'interaction.

Dans un système à initiatives mixtes, le système ne partage pas le même objectif que l'utilisateur, il assure la suppléance de l'humain qui réalise une tâche. Pour cela, le système surveille l'activité de l'utilisateur en s'appuyant sur des modélisations de celui-ci. Lorsque le système détecte qu'il peut aider ponctuellement à la résolution en cours du problème, il peut proposer son aide ou bien planifier son intervention si celle-ci peut être plus pertinente plus tard.

Systèmes malléables G. Fischer [94] enrichit des systèmes à base de connaissances (pour la résolution de problèmes) à l'aide de modèles de l'utilisateur et du système pour mettre en place une coopération. Le modèle de l'utilisateur du système n'a pas pour objectif de traquer les actions de l'utilisateur comme l'approche "initiative mixte", mais (1) de personnaliser les explications que le système peut donner, (2) de conserver des descriptions concurrentes des concepts utilisés et (3) personnaliser la présentation des informations. De manière symétrique, l'utilisateur se fait son propre modèle de la machine.

Pour l'auteur, une situation coopérative envisagée étant dynamique, le modèle de l'utilisateur qui sert aux schémas de coopération doit être malléable.

Dans la notion de système malléable [94], le système en interaction avec l'utilisateur doit à la fois être adaptable et adaptatif. Un système adaptable permet à son utilisateur de modifier à sa guise certains comportements du système (*e.g.* les systèmes comme l'éditeur de texte EMACS peut être paramétré pour les besoins d'un utilisateur). Un système adaptatif évolue de lui-même à partir des interactions avec l'utilisateur pour mieux correspondre à ses attentes futures. Pour Fischer, il est difficile de mettre initialement *toute* la connaissance utile à un problème dans un système. Il faut mieux envisager la situation où l'utilisateur complète les connaissances du système lorsqu'une résolution du problème est jugée inadéquate par l'utilisateur.

En résumé, on peut indiquer que les méthodes de collaboration homme/machine se caractérisent en se positionnant sur trois critères :

- l'échange d'explications : Le système est-il capable d'apporter une explication de ses actions ?
- le partage de l'initiative : Le système est-il capable d'être à l'origine d'une interaction avec l'utilisateur ou de l'allocation des tâches ?
- l'adaptation de la machine au cours de la collaboration : Le système évolue-t-il au cours de la collaboration ?

Dans notre approche, nous avons choisi une collaboration dans laquelle nous cherchons à partager à égalité l'initiative de l'interaction et de la proposition de solutions. Nous cherchons également à concevoir une machine s'adaptant fortement au cours de la collaboration. En revanche, nous avons pas choisi de faire échanger des explications entre les partenaires. Dans le cadre de la construction d'une interprétation, l'obstacle de la compréhension mutuelle d'une explication nous est apparu comme un problème important.

C.1.2 Méthodes d'échange de connaissances pour la compréhension mutuelle

Indépendamment de l'approche utilisée pour concevoir le système collaborant avec un utilisateur, l'échange des informations ou des connaissances entre les partenaires peut se faire de différentes manières. La problématique soulevée par Willamowski [223] est de trouver un niveau adéquate de communication entre le système et l'utilisateur de sorte à assurer une compréhension mutuelle (*cf.* figure 2.8).

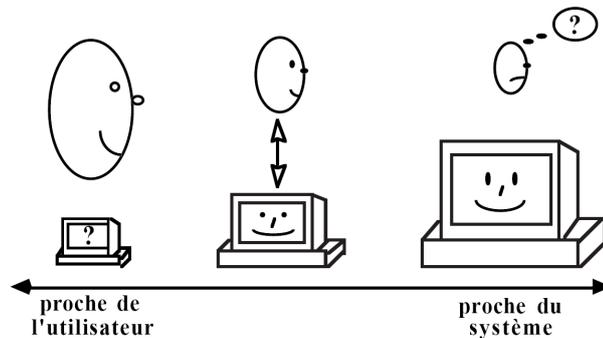


FIG. 2.8.

Choix d'un niveau de communication entre la machine et l'utilisateur d'après [223]. Sur la gauche, la communication est proche de l'utilisateur, la machine ne peut manipuler les données, sur la droite la communication est orientée vers la machine, mais l'utilisateur ne peut la comprendre. Au milieu, le compromis permet l'échange de connaissances.

Actes de langages Les ACA (Agents Conversationnels Animés) sont des agents virtuels généralement à l'apparence humaine. Ils peuvent assurer une interface entre un service et un utilisateur. Dans l'application de Chicoisne et Pesty [58], un ACA permet à un utilisateur à trouver un film dans une vidéothèque. Les interactions entre les agents permettent une résolution collaborative de ce problème. L'objectif des ACA est d'assurer un service convivial, facile d'utilisation et qui mette les utilisateurs en confiance. Certains ACA permettent des interactions multimodales, mais nous nous intéressons ici au travail qui a été développé dans ce contexte pour permettre des communications homme/machine au moyen d'un langage naturel [24]. Le langage naturel est ici écrit dans une interface et non parlé.

Bien évidemment, le langage est un moyen de communication avec lequel un humain est à l'aise, mais la difficulté est de permettre de rendre le contenu du langage accessible à la machine. Pour cela, les théories d'actes de langage (*e.g.* Austin [12] ou Vanderveken [50]) permettent de systématiser l'interprétation des textes et de générer des dialogues complexes entre la machine et l'humain pour répondre à ses besoins.

Explications et critiques Les études de dialogues coopératifs inter-humains de L. Karsenty et P. Brézillon [127] montrent que la résolution collaborative d'un problème revient non pas à trouver une solution, mais plutôt à trouver une *solution mutuellement acceptable*. Et par conséquent, la compréhension partagée du problème repose sur deux points : (i) l'apparition de conflits entre les propositions ou les jugements des différents partenaires, (ii) un processus explicatif qui conduit à construire un contexte de connaissances mutuellement acceptables permettant de supprimer le conflit. Les auteurs proposent donc une collaboration fondée sur l'échange de critiques et d'explications. De même pour Fischer [94], la critique et l'explication sont des composants essentiels à la résolution collaborative de problèmes.

Cependant, il apparaît dans Karsenty et Brézillon [127] que si les explications peuvent être générées par les machines, les explications fournies par les humains ne sont utilisées que comme connaissance brute par les machines. Horvitz [118] se contente pour sa part de la notion d'explication pour assurer une meilleure compréhension par l'utilisateur des propositions de la machine.

Communication par tâche Selon Willamowski [223], la notion de tâche comme modélisation du processus de la résolution d'un problème par décomposition récursive (ou planification hiérarchique) peut être appréhendée aussi bien par la machine que par un humain. Par conséquent, cet élément est jugé adéquat pour assurer les communications d'une collaboration.

La collaboration entre les deux agents est alors réduite à l'allocation collaborative des tâches que chacun aura à réaliser individuellement. Il est ainsi construit un plan de résolution dans lesquels les deux agents interviennent. Mais au cours de la résolution effective des tâches, il n'y a pas d'interactions entre les agents. De plus, l'établissement du plan est préalable à la résolution effective du problème. En particulier, il n'y a pas d'interactions pour négocier dynamiquement de nouveaux plans de résolution du problème.

Communication au travers d'un environnement partagé Une machine peut également avoir une interface physique, et non plus seulement une interface sur un écran. Les interactions qui peuvent être réalisées entre la machine et l'humain peuvent alors être très riches. Dans le domaine des CSCW, par exemple dans des applications à l'urbanisme [8], l'instrumentation de pièces ou d'outils permet à une machine de percevoir l'environnement de travail collaboratif et au moyen d'effecteurs tels que des vidéo-projecteurs d'augmenter et faciliter la collaboration entre les participants humains. Cependant, les situations de collaboration riche entre humains et machines dans un monde physique restent rares. Ceci est principalement lié au fait qu'il existe peu d'effecteurs physiques à disposition.

Le Continuator de F. Pachet [166] est un compagnon improvisateur pour des pianistes qui illustre une forme de collaboration dans le monde physique. Le musicien et le système sont tous les deux embarqués dans une même tâche "collaborative" d'improvisation musicale. Le système dispose d'un mécanisme d'apprentissage à partir de ce que joue le musicien pour lui-même improviser. Successivement, les deux agents échangent des mélodies qui leur permettent d'enrichir la suite de leur improvisation. D'un point de vue extérieur, tout semble comme si le système et le pianiste jouaient ensemble.

Échange d'annotations Il nous semble que le contenu d'une annotation peut être appréhendé par la machine comme par un humain. En effet, lorsque les contenus informationnels sont complexes, *e.g.* des images, il est difficile pour une machine d'en extraire le contenu et l'utilisation d'annotations permettrait à la machine d'avoir plus facilement accès à des informations riches.

Mais nous pouvons ici faire le même constat que pour la collaboration homme/machine dans un environnement physique, les annotations sont un moyen de collaboration entre les humains. Dans un contexte médical, les annotations de dossiers de patients permet la collaboration asynchrone au sein d'une équipe de soins [40]. Mais à notre connaissance il n'est pas utilisé comme moyen de communication entre machines et humains.

En résumé, l'échange d'informations entre un système informatique et un humain est aussi essentiel à la collaboration qu'il est difficile de le mettre en place. Ce qui caractérise l'échange des informations

dans une collaboration est selon nous :

- Une orientation vers les modes de communications humains ou plutôt d’un système,
- Le type d’objets qui sont échangés : actes de langages, tâches, explications, environnement, annotations.
- La symétrie des interactions.

La communication entre les agents peut être symétrique ou non, *i.e.* qu’il existe un premier mode d’échange de l’utilisateur vers le système et un second du système vers l’utilisateur. Les ACA les utilisateurs écrivent un texte et le système peut répondre vocalement. Outre cet exemple, la collaboration consiste souvent à trouver un mode de communication dans lequel un des agents soit à l’aise à la fois pour sa compréhension et pour sa production. À notre connaissance, il existe peu d’approches de communication explicitement dissymétriques. L’utilisation de communications symétriques peut être vu comme un biais de simplification introduit à la conception du système. D’un autre côté, cette symétrie permet de renforcer l’idée de non supervision du système par l’agent humain. En fait, les travaux récents sur les communications homme/machine se dirigent vers la multimodalité des interactions. Ces travaux visent ainsi à s’approcher de nos communications réelles entre personnes pour en faciliter l’utilisation intuitive.

C.2 Couplage Homme/Machine : nouveau paradigme de l’Enaction

L’exploration de la notion de collaboration nous conduit à considérer la machine suppléant l’humain non pas comme un simple outil auquel on imposerait les schémas cognitifs humains, ni même un système conçu pour reproduire ces schémas mais plutôt comme une entité autonome travaillant dans son propre espace de connaissance et de compétence.

Dans l’objectif de réalisation d’un tel système, nous nous sommes intéressés à le paradigme cognitive de l’Enaction (ou cognition incarnée) proposée par Varela [218] pour mettre en perspective la collaboration homme/machine. Ce paradigme est une alternative aux approches cognitives classiques (émergentiste ou computationnaliste). C’est l’étude des systèmes biologiques qui conduit Maturana et Varela [147] à proposer deux concepts clés :

- le concept de couplage structurel : Il s’agit d’un processus à travers lequel les structures de deux systèmes plastiques sont transformées par leurs interactions mutuelles.
- le concept système Enactif : Il s’agit d’un système qui construit son comportement au fur et à mesure de ses actions incarnées (*i.e.* de leur expérience propre).

Nous détaillons dans la sous-section suivante ces deux concepts, puis nous proposons une approche de la collaboration homme/machine comme un couplage structurel entre un homme et une machine. Le couplage structurel impose que l’humain et la machine soient des systèmes Enactifs, ce qui nous donnera un cadre de conception d’une machine pouvant être couplée avec un utilisateur humain (dont la nature Enactive est considérée vraie). Mais la notion d’expérience propre à chaque agent contraint la compréhension mutuelle. Nous explorons alors le champ des annotations comme média d’échanges d’informations et de connaissances entre la machine et l’utilisateur.

C.2.1 Couplage structurel et Enaction

L’œuvre de Varela est riche et la compréhension du concept d’Enaction, encore appelé paradigme Constructiviste, nécessite d’avoir une vision globale de sa pensée. Je tente ici modestement de faire la difficile condensation de son travail qui aura en partie inspiré cette thèse. Duquaire [90] propose une introduction à la pensée de Varela qui donnera plus de détails philosophiques sur l’évolution de sa pensée. Le lecteur pourra également s’enrichir des enseignements de J. Theureau dans le cadre de ses cours sur les théories et méthodes d’analyse de l’action¹³.

Francisco Varela s’est d’abord intéressé conjointement avec Humberto Maturana aux systèmes vivants et à leur autonomie. Un système vivant est une “machine” qui se caractérise par deux notions duales (1) sa structure, *i.e.* les composants du système, et (2) son organisation, *i.e.* les relations entre les composants. Les caractéristiques d’un système vivant selon Varela et Maturana sont regroupées dans la notion de système autopoïétique.

Ethymologiquement autopoïèse renvoie à *autos*, soi, et *poiein*, produire. Un **système autopoïétique** est donc un système qui la capacité de se produire lui-même. Ceci est à l’image d’un organisme vivant

¹³Theureau J. (1999), Supports de cours de l’UTC/SHT accessibles sur internet <http://www.coursdaction.net/>.

dont les mécanismes internes permettent de maintenir sa structure. Plus simplement une cellule est un bel exemple d'organisation autopoïétique. Elle se compose d'une membrane, qui la sépare du milieu extérieur et renferme un ensemble de composants interagissants continuellement en réseau (réseau d'interactions moléculaires). La membrane participe à ces interactions au même titre que les autres composants et contribue à la production de la cellule tout entière. Pour une illustration formelle de la notion d'autopoïèse, Bourguin et Stewart [33] proposent un automate de Tessellation comme modèle d'un système autopoïétique minimal à l'image d'une cellule.

Dans les termes de Varela et Maturana ([216], p. 45), un système autopoïétique est "organisé comme un réseau de processus de production de composants qui (a) régénèrent continuellement par leurs transformations et leurs interactions le réseau qui les a produits, et qui (b) constituent le système en tant qu'unité concrète dans l'espace où il existe, en spécifiant le domaine topologique où il se réalise comme réseau".

Les systèmes autopoïétiques sont des systèmes autonomes dans le sens où ils sont capables de maintenir leur organisation et leur frontière malgré les perturbations qu'ils peuvent subir de leur environnement (*i.e.* tout ce qui est hors de ses limites).

Deux notions importantes des systèmes autopoïétiques (donc autonomes) soulèvent des difficultés non résolues, d'une part la frontière (*e.g.* la membrane de la cellule) qui définit le système et en particulier la séparation du système avec son environnement (*e.g.* pour certaines interactions moléculaires le choix entre interaction auto-constructive ou perturbation extérieure peut être ambiguë). La seconde notion problématique est celle d'unicité. Dans le cas d'une cellule, le système est en constante reconstruction et ses composants (moléculaires) sont changés quasiment intégralement sur une durée très inférieure à sa durée de vie. Pourtant, on s'accordera à dire qu'il s'agit de la même cellule. De manière générale, ce critère qui permet d'indiquer la conservation de l'unité n'est pas facile à définir.

La notion de système autopoïétique trouve sa place dans l'étude des systèmes vivants, mais est généralisée pour une plus grande variété de systèmes autonomes (telles que des sociétés humaines étudiées par Luhmann) par la notion de **clôture opérationnelle**. "Un système autonome est opérationnellement clos si son organisation est caractérisée par des processus (a) dépendant récursivement les uns des autres pour la génération et la réalisation des processus eux-mêmes et (b) constituant le système comme unité reconnaissable (le domaine) où les processus existent" ([216], p. 86). Un système opérationnellement clos ne dispose pas d'entrée ni de sortie (le système est clos), il est soumis à des perturbations, des interférences de son environnement qui le conduisent à modifier sa structure de sorte à maintenir son organisation et donc, du point de vue de l'Enaction, son autonomie. La thèse défendue alors par Varela est que tout système autonome est opérationnellement clos car c'est un mécanisme qui est intimement lié avec la conservation de l'identité d'un système.

Le système autonome subit des perturbations de son environnement qui le pousse à déformer sa structure (de sorte à maintenir son organisation). Varela et Maturana distinguent deux sources de déformation : les déformations qui proviennent de l'environnement d'origine indépendante au système et celles qui proviennent du système lui-même en tant que compensation d'une déformation déjà subie. Dans le processus autopoïétique, il est impossible à un observateur extérieur de distinguer l'origine d'une déformation entre ces deux sources. Et c'est pourquoi, le système et son environnement sont intimement liés.

Tout au long de son histoire, le système évolue de sorte à ce que sa structure plastique soit déterminée par les perturbations récurrentes de l'environnement. Cette ontogenèse du système constitue un parcours unique de la structure du système. La structure caractérisant le *comportement* du système, celui-ci dépend également de l'ontogenèse. On peut ajouter qu'un système autonome est non-téléologique d'après Maturana. C'est-à-dire que celui-ci n'a pas de but fixé et donc n'a pas de fin prévue. La seule préoccupation du système est celle de son fonctionnement interne d'adaptation de sa structure aux perturbations extérieures pour la conservation de son unité.

Si on suppose que l'environnement est lui-même un système autopoïétique, alors les deux systèmes se répondent au travers de perturbations qu'ils s'imposent par leurs déformations structurelles réciproques. Ce processus est nommé par Maturana **couplage structurel**. En fait, Maturana et Varela distinguent trois sortes de couplages structurels ([148], p. 88) :

- Un couplage de premier ordre entre une unité autopoïétique et son environnement,
- Un couplage structurel de second ordre entre un organisme ayant un système nerveux et le système nerveux lui-même,
- Un couplage structurel de troisième ordre entre de tels organismes dotés de systèmes nerveux.

Dans la suite, seule la notion de couplage structurel se rapportera à celui du troisième ordre uniquement.

L'une des implications de la modification structurelle d'un système est la modification de son domaine phénoménal. C'est-à-dire de l'ensemble des phénomènes (au sens de la phénoménologie de Husserl ou Merleau-Ponty) qui peuvent être acceptés comme déformations futures. Ce qui conduit une unité systémique à "spécifier un domaine avec lequel elle peut interagir sans perdre ni sa clôture ni son identité" ([216], p. 88).

Le fondement de la pensée de Varela sur la cognition se trouve dans l'intuition de Maturana de concevoir le système nerveux comme un système autonome, opérationnellement clos. Il en découle une approche de la cognition qui contredit les acceptations usuelles qui sont faites dans le domaine des neurosciences (aussi bien le connexionisme que le cognitivisme) qui consistent à voir le système nerveux comme un système de traitement d'informations constitué d'entrées et de sorties.

Le système nerveux étant une entité autonome, le paradigme de l'**Enaction**¹⁴ indique que le "soi" est construit par les expériences dans le monde et il nie la tradition cartésienne de la dualité de l'esprit et du corps.

Dans son invitation aux sciences cognitives [217], Varela situe l'Enaction comme une voie alternative au connexionisme et au cognitivisme. Le connexionisme de Minsky[154], Hofstadter [115] ou Changeux [55] définit la cognition comme étant une propriété émergente des interactions entre des neurones localement interconnectés. La théorie connexionniste ancre une représentation du monde dans la structure du réseau de neurones. À l'opposé, le cognitivisme défendu par Fodor [95] ou Chomsky définit l'esprit comme une machine fondamentalement symbolique qui fonctionne à partir de règles formelles.

Varela et *al.* défendent l'idée d'une cognition qui n'est pas faite au moyen de représentations du monde, mais d'une expérience vécue, incarnée dans un environnement physique. Les notions de "représentation", de "stockage" et même d'"information" sont abandonnées au profit des notions de "construction" interne (d'où le nom également de paradigme constructiviste) et d'"in-formation" (valant pour "formation interne").

Sans notion d'entrée et de sortie dans la cognition, l'approche "Enactive" met au centre de la cognition l'incarnation du corps dans son environnement. Dans la suite de la notion de couplage structurel, un système cognitif est le résultat d'un couplage sensori-moteur fait d'actions et de perceptions dans l'environnement, et même plus précisément d'actions puisque la perception est vue comme guidée par l'action. En effet, le concept d'Enaction est défini par Varela, Thompson et Rosch comme une approche qui se compose en deux points : "(1) la perception consiste en une action guidée par la perception ; (2) les structures cognitives émergent des schèmes sensori-moteurs récurrents qui permettent à l'action d'être guidée par la perception" ([218], p.234).

Tout d'abord, pour Varela et *al.*, la perception est guidée par l'action. Pour mettre en évidence cela, Varela relate l'expérimentation de Held et Hein [111] qui consista à constituer deux groupes de chatons. Les chatons du premier groupe étaient libres de tout mouvement si ce n'est que chacun était attelé avec un chariot qui embarquait un autre chaton appartenant au second groupe. L'idée étant que les chatons du premier groupe et du second groupe vivaient la même expérience visuelle, mais que seul les animaux du premier groupe étaient actifs dans leur environnement. Après quelques semaines à subir ce traitement, l'ensemble des chatons étaient relâchés librement, tandis que les chatons se comportaient normalement, les chatons du second groupe se cognaient partout "comme s'ils étaient aveugles" ([218], p. 237). Cet exemple met en évidence l'importance de l'expérience vécue qui se retrouve dans le paradigme constructiviste qui indique qu'un savoir-faire ne peut s'acquérir que par la pratique et non pas uniquement de manière théorique. On pourra par exemple penser à un sportif qui ne doit pas se contenter de regarder les autres sportifs pour savoir réaliser un geste. Il en est de même de la perception, l'extraction des éléments saillants¹⁵ nécessite une pratique. Lorsqu'un radiologue doit interpréter une radiographie, il s'appuie sur son expérience pour être capable d'identifier une fracture qu'un quidam ne pourrait *percevoir*.

Ces deux derniers exemples sont à rapprocher d'une conception Gibsonienne de l'environnement. La

¹⁴ La première utilisation du terme Enaction a été faite par Bruner (1966) [41] comme la proposition d'une alternative à une conception de l'esprit iconique ou symbolique. Le terme est utilisé par Varela et *al.* pour parler de *cognition incarnée*.

¹⁵ Nous adoptons ici une conception très objectiviste de la perception.

théorie de Gibson [102] est une théorie des affordances (voir également Uexküll [215]). Pour Varela, les affordances étant des informations préexistantes du monde qu'un système cognitif doit identifier, l'approche écologique de la cognition est une approche trop objectiviste. Dans la tradition phénoménologique de Merleau-Ponty, "l'organisme donne forme à son environnement en même temps qu'il est façonné par lui" ([218], p. 236). Varela note sur ce point la diversité des solutions "viabiles" trouvées dans la nature pour le problème de la perception visuelle (*e.g.* trichromique ou tetrachromique). Le paradigme de l'Enaction empreinte beaucoup la phénoménologie de Husserl ou Merleau Ponty pour lesquels l'expérience constitue également un fondement de la construction non-objective du "soi".

C'est tout le sens du couplage structurel avec l'environnement de préciser que les structures cognitives émergent par des schèmes sensori-moteurs mais également qu'un monde émerge de cette construction. Le système et l'environnement sont ainsi co-déterminés au long de l'expérience vécue par le système cognitif.

Selon Merleau-Ponty, il est impossible de connaître un objet comme il est lui-même. Il est seulement possible de connaître un objet selon ce qui nous est accessible de lui. Un système cognitif ne peut concevoir un objet qu'à travers des moyens qui imposent des limites à sa perception. De plus, par le processus de couplage structurel avec l'environnement le phénomène existe pour un être car il traduit une intention d'un objet pour lui.

Par exemple, à leur naissance, les bébés ont des capacités étendues de distinction de sonorités, et c'est par leur environnement sonore que se réduit le champ des sons qui pourront être distingués et prononcés à l'âge adulte et qui rend difficile l'apprentissage de l'arabe aux français et réciproquement. Dans sa construction ontologique, le bébé se construit un monde de sons et qui lui est propre.

L'expérience vécue étant individuelle et déterminante pour l'internalisation du monde, de l'environnement, la conséquence du paradigme de Varela est le subjectivisme de la conception du monde.

En résumé, le paradigme de l'Enaction nous amène à penser un système cognitif comme un système autonome (autopoïétique et donc par essence clos et plastique) et incarné. La construction de ce système, dépendant de ses capacités physiques, se fait par un couplage structurel avec son environnement. C'est-à-dire que ce système (le "soi") et l'environnement (le "non-soi") se co-déterminent dans un processus sans fin d'interactions réciproques. Les schèmes sensori-moteurs sont eux-mêmes déterminés par l'historique du couplage.

C.2.2 Enaction et informatique

Le paradigme de l'Enaction fait maintenant son entrée dans les technologies de l'information et de la communication¹⁶ en particulier pour renouveler le domaine des interfaces Homme/Machine en proposant des modes d'interaction qui ne soient ni iconiques, ni-symboliques mais incarnés¹⁷.

Les interfaces Haptiques [110] permettent des interactions avec le corps d'un utilisateur. Par exemple, les stylets ou les gants à retour de forces qui permettent à des modeleurs en trois dimensions de "sentir" la matière modelée. On retrouve ici l'idée de couplage structurel grâce au retour de l'environnement qui est possible par l'interface haptique alors que la simple souris ne le permettait pas.

Les interfaces de suppléances perceptives¹⁸ sont également un domaine dans lequel des applications prometteuses sont envisagées. Ces interfaces sont des sortes de prothèses sensorielles.

L'opérationnalisation du concept d'Enaction entreprise par Dionisi [80] vise à dépasser la conception des programmes informatiques comme "de simples instruments utilitaires soumis à des buts pré-établis" [109] en caractérisant les "processus logiciels" comme impliqués dans des "processus expérimentiels impliquant des processus cognitifs" (*i.e.* un utilisateur humain) [81]. Pour cela, la démarche entreprise consiste à favoriser l'intégration du processus logiciel à l'action cognitive de l'utilisateur en le pensant comme un

¹⁶On retrouve également le paradigme de l'Enaction dans une approche constructiviste de l'étude de comportements sociaux (*e.g.* en situation de travail, en situation d'enseignement, en situation de crise). En particulier, dans les théories de l'action située [207], théorie du cours de l'action [208] ou de la cognition distribuée [120] qui donnent des approches de conception lorsque des aspects humains sont à prendre en compte.

¹⁷Le réseau d'excellence Européen "Enactive" regroupe les acteurs des interfaces dites Enactives.

¹⁸La suppléance perceptive est inspirée des expériences de Bach y Rita en 1969 [14] qui utilisa des caméras et des dispositifs vibrants sur la peau pour constater l'acquisition d'une perception de l'environnement par un apprentissage.

couplage entre l'utilisateur et la machine plutôt que la démarche inverse utilisée classiquement en IHM.

L'opérationnalisation de l'Enaction conduit Dionisi à proposer une conception d'un système informatique autonome et non-téléologique. Pour ne pas tomber dans une interaction sous la forme d'entrées et sorties, les interactions sont constantes entre l'utilisateur et le logiciel. Il n'y a pas de communication directe logiciel/utilisateur mais tout s'effectue par des perturbations de l'univers commun dont le seul point d'appui repose sur le domaine de connaissance qui doit être abordé par le système. La prise en compte des perturbations du milieu ambiant s'effectue par le mécanisme d'internalisation, de "mise en cohérence des processus". Cette approche a été appliquée à un logiciel de jeu d'échec internalisant les contraintes du jeu d'échec.

C.2.3 Une nouvelle approche de la collaboration Homme/Machine

Nous proposons ici une nouvelle approche de la collaboration homme/machine à partir du paradigme de l'Enaction présenté dans la section précédente. L'objectif est de mettre en place une collaboration entre l'homme et la machine à l'image d'un couplage structurel entre les deux agents. Par rapport à la troisième assomption du couplage structurel mettant en connexion directement deux organismes dotés de systèmes nerveux, nous utilisons une variante consistant à prendre un environnement partagé comme intermédiaire (donc des couplages du premier ordre). Chaque agent est alors structurellement couplé à l'environnement de sorte que les actions qui y sont faites soient indirectement perçues comme une modification de l'environnement par l'autre agent. L'action dans l'environnement peut être réalisée dans l'intention de transmettre un message à l'autre ou bien la modification de l'environnement peut être une action réalisée pour elle-même. Auquel cas, la prise en compte par l'autre agent est alors un *effet de bord*.

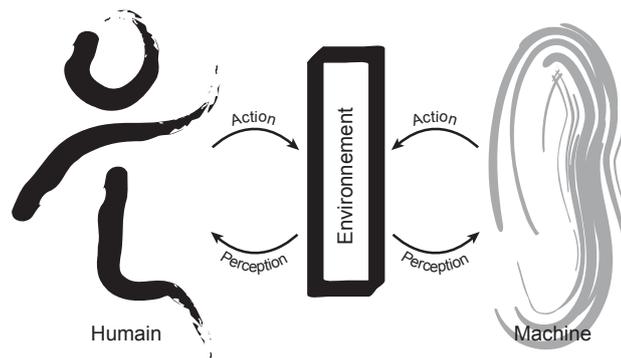


FIG. 2.9.

Schéma de principe de la collaboration entre la machine et l'humain au travers d'un environnement. Les deux boucles action/perception figurent des couplages structurels entre chacun des agents et l'environnement.

Nous proposons une situation de collaboration dans laquelle deux agents, l'homme et la machine sont considérés comme autonomes et que ceux-ci collaborent dans le but de réaliser une même tâche. En conservant leur autonomie, les agents conservent leurs spécificités de traitement. Dans le cadre d'une résolution de problème, nous cherchons à profiter de ces spécificités pour disposer de deux vues différentes du même problème pour enrichir les voies possibles de résolution. Donc chacun des deux agents, ancré dans son propre espace cognitif, réalise en parallèle cette même tâche.

Au travers du couplage structurel entre la machine et l'homme, les deux agents suivent un parcours congruent qui les amène à la résolution partagée du problème. Les deux systèmes autonomes s'adaptant l'un à l'autre, d'une part les performances du couple à réaliser la tâche sont améliorées au fur et à mesure de leurs expériences, mais d'autre part chacun profite des interactions pour raffiner ses propres compétences et perceptions. La machine et l'homme n'étant pas sensibles aux mêmes phénomènes, on cherche au travers du couplage à faire en sorte qu'ils puissent se rendre sensibles ou plutôt *attentifs* à de nouveaux phénomènes. On préférera dire qu'un agent devient plus attentif, car en particulier pour la machine sa perception est limitée et très contrainte physiquement. Ceci est clairement une limite importante pour reconnaître dans la collaboration homme/machine un véritable couplage structurel.

La collaboration intervient entre les deux agents comme des perturbations de l'un sur l'autre. Ces perturbations orientent un agent dans sa résolution du problème. La perturbation d'un agent vise à faire

réaliser une tâche par l'autre agent de sorte à ce que ce dernier intègre (dans son processus de constante évolution) de nouvelle façon de résoudre un problème, charge à chaque entité autonome de savoir si avec ses compétences propres, la nouvelle voie proposée est adéquate ou non. Il est central dans cette approche de mettre beaucoup d'efforts à conserver les agents autonomes dans leur acquisition de l'"in-formation".

La machine permettant cette collaboration doit être :

- capable de résoudre la tâche ou des sous-tâches de manière autonome,
- hautement adaptative à partir de sa propre expérience,
- capable d'interagir en préservant son autonomie.

Ce troisième point nous amène à proposer un mode d'interaction permettant la compréhension mutuelle tout en respectant l'autonomie des deux agents collaborant.

C.2.4 Annotations comme échange de connaissances dans les systèmes couplés

Willamowski [223] pose le problème de l'interaction entre un humain et une machine comme un problème de choix d'un niveau de description adéquate qui permettent une compréhension mutuelle des deux parties par la communication de ce type de description. Du point de vue du style de collaboration que nous avons mis en place, cette même démarche peut se traduire comme la recherche d'un environnement qui pourra servir de médium pour les deux agents, *i.e.* qui pourra être perçu et modifié par chaque agent.

L'environnement comme support à la communication a déjà été utilisé dans la communication entre les *Talking Heads* de Steels et *al.* [203] déjà présentée dans la section A.2. L'originalité du mode de communication tient à la fois dans le fait de communiquer au travers d'un environnement partagé par deux *têtes parlantes*, les formes situées sur un tableau blanc, mais aussi dans l'information qui est échangée lors de communication à savoir des instances des concepts plutôt que les concepts eux-mêmes. Et c'est ce second point qui nous intéresse particulièrement ici.

Les interactions sont basées sur des jeux de langage, des "énigmes" (*Guessing games*). Une "énigme" est jouée par deux agents, celui qui pose l'énigme et celui qui la résout. L'agent qui pose l'énigme choisit un objet parmi les objets du tableau blanc, et il communique ensuite à l'autre agent le nom qu'il attribut à cet objet, selon son propre langage. Le second agent, qui cherche à comprendre ce qui a été dit par le premier. À partir de l'hypothèse qu'il formule, il communique à l'autre agent un exemple dans l'environnement partagé qu'il pense correspondre à cette définition. Si l'objet pointé correspond à l'objet initial pour l'agent questionneur, le jeu est un succès, sinon les agents raffinent leur langage pour éviter une erreur la fois suivante.

Les expérimentations montrent qu'avec un mode d'interaction qui est basé sur l'échange d'instances (éléments du langage et des objets) et non sur l'échange de concepts (comme par exemple la description du langage sous la forme de correspondances des *mots* aux objets), des connaissances peuvent être échangées de sorte à construire le langage partagé.

L'échange d'objets appartenant à l'environnement partagé nous apparaît être une solution viable pour la collaboration que nous souhaitons mettre en place. Les expérimentations des *Talking Heads* montrent les réelles possibilités d'échange de connaissance qui peuvent être espérées et du point de vue d'une conception d'un couplage structurel entre un système et une machine. Cette solution permet d'assurer une autonomie des deux agents. Contrairement à Steels et *al.* qui cherchent à faire émerger un langage commun, nous souhaitons pour notre part que chaque agent reste dans son *espace de sens* propre. Nous mettons en évidence ce point grâce à la notion d'annotation qui constituent un type d'objets tangibles qui peut assurer la communication au travers de l'environnement.

Les annotations en tant qu'objets qui peuvent être échangés dans l'environnement constituent des médiums de communication pour une collaboration à l'image d'un couplage structurel. Dans un contexte hospitalier, Bringay [40] estime que les annotations de documents médicaux sont l'un des éléments essentiels pour l'élaboration et le maintien de la collaboration au sein d'une équipe médicale. Et dans des applications industrielles, les annotations sont également largement utilisées comme aide à la conception collaborative [10, 32].

Bringay [40] fait une étude fine du concept d'annotation en considérant l'annotation comme un acte de langage. Ceci permet de voir le contenu de l'information selon 4 dimensions :

- La dimension locutoire/illocutoire : Une annotation est définie par Bringay [38] comme une “note particulière liée à une cible par une ancre”. La cible peut être une collection de documents, un document, une partie de document ou encore une autre annotation. L’ancre lie l’annotation à la cible, *e.g.* texte surligné ou un *post-it* collé.
- La dimension contextuelle : L’annotation est à la fois écrite dans un contexte particulier à l’annotateur et lue dans un autre par son lecteur (qui n’est pas nécessairement le même).
- La dimension perlocutoire : L’annotation sert pour son auteur pour se rappeler, penser, clarifier ou enregistrer une réflexion incidente. Pour le lecteur, elle sert à retrouver l’information ou sa structure, faciliter la compréhension, réinterpréter le document.
- La dimension collaborative : On a déjà vu dans un paragraphe précédent que l’annotation est un élément pour faciliter la collaboration.

Le modèle d’annotation des dossiers patients informatisés qui en découle [39] doit être adapté à une situation de collaboration Homme/Machine. En particulier, le contenu textuel d’une “note” ne peut être utilisé puisque c’est une information que les machines peuvent difficilement extraire. Nous proposons en section D.3.1 une sémantique annotationnelle [16] spécifique à notre application.

L’annotation s’inscrit également dans le paradigme de l’Enaction, car c’est un *acte* de (re-)création de sens pour les autres, mais également pour soi-même. Et ce sens modèle également le contenu d’un document en y apportant des éléments nouveaux à interpréter pour les lecteurs suivants. De plus, l’annotation ne doit être seulement vue comme une intention de communication, elle est aussi un moyen de recréation d’un contenu et donc de son appropriation. D’après Bachimont [15], “en y apportant sa propre marque, le lecteur devient l’auteur de sa propre lecture dans la mesure où il reformule dans ses propres termes ce qu’il voit : en quelque sorte, il écrit le document qu’il aurait voulu consulter s’il avait été l’auteur”. Cette appropriation du contenu du document qu’impose la communication par les annotations participe pleinement au bon déroulement d’un couplage structurel dans lequel le sens peut être partagé mais qu’il est ancré dans les *espaces des sens* de chaque agent.

D Interprétation collaborative

Une tâche d’interprétation peut bénéficier d’une collaboration entre agents hétérogènes comme le sont un homme et une machine. En effet, on a pu voir que les connaissances préalables sont déterminantes pour le résultat de l’interprétation. L’approche Enactive de la cognition nous indique également que les capacités perceptives et cognitives de l’agent la déterminent pour partie. Les trois facettes (connaissances, capacités perceptives et capacités cognitives) définissent pour nous le point de vue d’un agent sur les données. Les points de vue de l’homme et la machine sont différents. Faire collaborer ces agents hétérogènes doit permettre de les confronter de sorte à en faire émerger un résultat bénéficiant de cette différence.

D.1 Interprétation collaborative à la lueur de l’Enaction

L’Enaction a été choisie comme approche théorique de la collaboration. L’interprétation telle que nous l’avons présentée partage la base phénoménologique du paradigme de Varela. Revoir l’interprétation à la lueur du paradigme de l’Enaction nous permet d’ancrer l’Enaction comme le point central de notre proposition théorique de l’interprétation collaborative.

D.1.1 Point de vue Enactif de la tâche d’interprétation

On peut faire des rapprochements forts entre les éléments de l’interprétation et les notions introduites par Varela et Maturana. Le tableau 2.1 illustre quelques rapprochements deux à deux à partir des quatre grandes caractéristiques présentées de l’interprétation.

Le parcours interprétatif a été présenté comme une évolution au cours de l’interprétation. À chaque étape de ce parcours, l’interpréteur s’enrichissait par des essais/erreurs qui constituaient une forme d’*expérience* de l’objet interprété. L’expérience passée d’un système Enactif et le parcours interprétatif étaient individuels et déterminant pour l’état présent du système ou ici de l’interprétation.

De plus, nous les pensons tous les deux comme des processus circulaires. Donc, il s’agit de deux processus potentiellement sans fin car toujours alimentés par de nouvelles informations à intégrer pour compléter l’expérience passée.

Interprétation	Enaction
Parcours interprétatif	Expérience
Objectif inatteignable de la cohérence	Non finalité des processus
Appropriation par la recréation	Approche constructiviste
Cohérence	Structure opérationnellement close

TAB. 2.1.

Mise en correspondance de l'interprétation et des concepts de l'Enaction

L'individualisation de l'interprétation dans ce parcours interprétatif passe en partie par la recréation par l'interpréteur de l'objet interprété. De même pour l'Enaction, également appelée approche constructiviste, la subjectivité de la conception du monde passe par sa reconstruction. L'appropriation par la recréation trouve donc sa correspondance dans l'Enaction.

La construction de la cohérence des éléments d'une interprétation apparaît comme la construction d'une structure opérationnellement close. Le processus de mise en cohérence a été introduit comme un processus d'émergence de sens pour l'interprétation. L'émergence telle que nous l'avons introduite et mise en œuvre sur l'émergence de sens tient d'un processus autopoïétique. Pour le système autopoïétique comme pour l'émergence, on cherche à construire une *structure* viable par une dynamique régulée sur plusieurs niveaux.

Il ne semble pas qu'il s'agisse ici d'une sur-interprétation d'un lien entre les deux, mais simplement d'une mise en évidence que, comme l'interprétation philologique le prétend, l'homme ne fait qu'*interpréter*, les schémas de l'interprétation se retrouvent dans ses schémas cognitifs.

Le rapprochement de l'interprétation avec le paradigme de l'Enaction fait de l'interprétation une *action incarnée* comme cela a déjà été mis en évidence par C. Garbay [100] dans le cadre d'un système de vision. Il nous semble intéressant de justifier ce rapprochement au regard de deux notions de l'Enaction : les notions d'action et d'environnement (dans lequel l'agent peut être incarné).

Nous avons mis en évidence à propos de notre approche de la collaboration que l'environnement devait permettre la mise en relation des deux agents (*cf.* figure 2.9). On prend alors la notion d'incarnation dans un sens très large qui indique qu'un système est incarné s'il peut modifier son environnement, au moyen d'actions, et également être modifié par lui. L'action étant alors ce qui permet à un agent de modifier l'environnement. En particulier, la réalisation d'une annotation est une action qui modifie la façon de percevoir l'environnement. Par exemple, les parties d'un document écrit préalablement surlignées seront lues avec plus d'attention que les autres parties. Ces mêmes parties surlignées permettent un parcours rapide du document s'appuyant uniquement sur ces contenus car on suppose qu'elles contiennent l'essentiel du document. Mais en modifiant notre parcours du document, on modifie l'interprétation générale que l'on va en faire. Par conséquent, les annotations peuvent s'inscrire dans des sortes de "schèmes sensi-moteur".

D.1.2 Annotations et interprétation

Le terme d'annotation utilisé pour évoquer la communication via un environnement prend tout son sens dans une tâche collaborative d'interprétation. Par exemple, dans le cadre de l'interprétation d'œuvre d'art, le système de Bottoni et *al.* [31] propose de faire collaborer des experts au moyen d'annotations qui vont permettre de collaborativement construire une interprétation d'une peinture. La lecture des annotations s'intègre à plusieurs niveaux dans le processus d'interprétation.

D'une part, les informations contenues dans une annotation peuvent être traitées quasiment au même titre que les autres données. J. Derrida [76] considère que les annotations sont des éléments intertextuels. Il s'agit de productions sémiotiques correspondant à un discours à propos d'un texte et que l'annotation a donc un sens via la relation qui l'unie au texte annoté. Contrairement aux données initialement à interpréter, ces annotations peuvent être critiquées, *i.e.* renforcées, modifiées ou niées.

L'annotation est également intrinsèquement porteuse de sens. C'est-à-dire qu'indépendamment d'un contenu sémantique additionnel, la marque de l'annotation est déjà porteuse de sens. D'une part, la marque délimite une partie d'un document à interpréter qui est particulièrement intéressant, *i.e.* dont le contenu informationnel doit être pris en considération. D'autre part, le type de marque est également parlant. Par exemple, pour un texte surligné, la présence de deux couleurs de surlignage différentes révèle deux natures différentes dans les informations surlignées (*e.g.* les verbes et les noms ou deux annotateurs

différents ou encore des annotations à propos de deux sujets différents ...). Dans ce cas, la sémantique n'est pas explicite.

Ensuite, les annotations peuvent servir à construire le parcours des données. En effet, les annotations peuvent être parcourues en tant que telles (*e.g.* parcourir toutes les parties surlignées par Mr.X d'un texte), mais dans la mesure où elles sont ancrées à des parties du document initial, elles offrent de nouvelles dimensions pour parcourir le document. Dans la section précédente, on a illustré ceci par la possibilité de faire un parcours rapide d'un document écrit à partir de ses parties surlignées.

Enfin, l'annotation en temps qu'action intervient également au moment de réaliser une rétro-action sur les données. L'"incarnation" d'une marque sur le document initial met en confrontation les données du document avec l'information qui va être portée par l'annotation.

D.1.3 Construction d'une cohérence entre le système et l'humain

Chacun des agents, le système et l'humain, doit construire ses propres connaissances cohérentes en interne et avec son expérience. Ceci implique également une cohérence avec les éléments de l'environnement qu'il peut percevoir. Sur la figure 2.10, cela revient à dire que les ensembles $\{Environnement + Homme\}$ et $\{Environnement + Systeme\}$ doivent être cohérents. Les deux systèmes ayant cette priorité, la collaboration entre le système et l'humain va conduire à une cohérence globale entre le système et l'humain. Cependant, cette cohérence n'implique que la non-contradiction des éléments d'annotation incarnés dans l'environnement. Des incohérences peuvent persister entre des éléments internes à l'homme et ceux du système. Ceci permet de conserver l'autonomie d'interprétation des deux agents.

En fait, nous avons indiqué que dans l'idéal une tâche d'interprétation est infinie parce qu'on peut toujours identifier quelque chose qui va perturber la cohérence globale du système. En pratique, la machine comme l'humain réalisent des interprétations qui sont finalisées. Les êtres humains ont la faculté d'identifier des éléments saillants qui pourront mettre en branle la cohérence déjà établie et donc à rendre l'interprétation infinie, mais ils ont également une capacité à prendre en compte des éléments marginaux. Lorsqu'il est nécessaire d'arriver à une cohérence, il est parfois nécessaire de mettre à l'écart certains éléments qui ne *semblent* pas rentrer dans la cohérence globale recherchée. Donc en pratique aussi, on peut compter sur les humains pour fournir une interprétation finalisée.

Quant au système, il n'a à disposition qu'une quantité limitée de données à interpréter. Par conséquent, il est possible d'envisager des processus qui finissent.

D.1.4 Annotation comme moyen d'échange dans la collaboration

Dans le processus global de collaboration, la construction de connaissances cohérentes permet de faire passer au système une partie de la connaissance du domaine spécifique au clinicien. Cette transmission est indirecte et non-formalisée : on a en effet évacué l'idée d'échanger des connaissances directement de l'un à l'autre. Mais dans la mesure où le système acquiert des capacités spécifiques au domaine (*e.g.* la capacité de distinguer des formes significatives, qui ne sont pas les mêmes pour des données d'un domaine applicatif à l'autre), il extrait des annotations de l'utilisateur une certaine connaissance du domaine.

Sur la figure 2.10, on peut voir qu'au travers des annotations, il peut se créer des liens indirects entre les connaissances d'un système et celles du clinicien. Les liens sont "indirects" dans le sens où il y a nécessairement une appropriation de l'annotation. Au travers de ces liens indirects, les agents vont mutuellement s'enrichir. En effet, l'adaptation des connaissances d'un agent déclenchée par l'appropriation des annotations lui fait construire des connaissances concordantes avec les annotations (et donc les connaissances de l'autre agent). L'utilisation d'annotations pour l'interprétation collaborative permet donc un partage des connaissances tout en préservant l'autonomie des interprétants.

D.2 Caractéristiques du système pour l'interprétation collaborative

Dans l'objectif de faire réaliser une interprétation collaborativement par la machine et le système, le système doit répondre à des caractéristiques spécifiques (que nous attribuons par défaut au clinicien). Nous devons d'une part mettre en place un couplage structurel entre le système et l'utilisateur, et d'autre part assurer une communication au travers d'annotations.

Pour assurer une communication au travers des annotations, notre système doit être capable d'intégrer

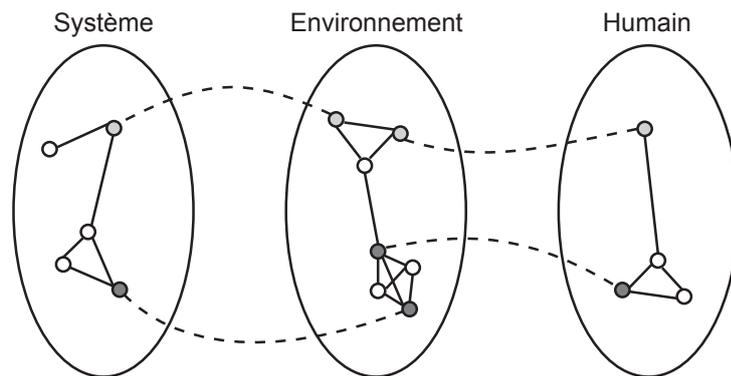


FIG. 2.10.

Illustration de la transmission des connaissances par “transitivité” des cohérences. Les liens en pointillés représentent des relations entre un des agents interpréteur et l’environnement, tandis que les liens pleins illustrent des relations entre les éléments internes d’une des trois structures.

les annotations du clinicien et produire ses propres annotations. La sémantique des annotations doit alors être suffisamment formalisée pour que ces annotations soient des connaissances qui peuvent être partagées par le système et l’interpréteur humain. Ceci ne contraint pas théoriquement notre système, mais la réalisation d’annotations *propres* pour permettre de conserver l’hétérogénéité des points de vue implique d’être capable de conserver l’autonomie du système pendant l’interprétation. Pour produire des annotations de manière autonome, nous devons limiter l’introduction de connaissances *a priori*.

Pour répondre aux besoins d’un couplage structurel, le système doit avoir les propriétés d’un système autopoïétique (cf. section C.2.1). La construction d’un système autopoïétique implique d’une part que le système soit plastique, *i.e.* un système dont les capacités “perceptives” et “cognitives” soient suffisamment adaptables pour qu’il puisse évoluer au cours de l’expérience du couplage. D’autre part, ceci implique que le système soit capable de produire sa propre cohérence.

On dégage ainsi trois requis forts que notre système doit respecter :

- Le système doit être capable de faire émerger une interprétation des séries temporelles (en incluant la mise en cohérence des annotations et des modèles).
- Il faut limiter l’introduction de connaissance *a priori* pour assurer l’autonomie d’interprétation du système.
- Le système doit adapter sa “perception” et sa “cognition”. Ceci revient à faire évoluer les modèles des données au cours de l’interprétation.

Ces caractéristiques nous ont guidés dans la conception du système qui sera présenté dans le chapitre 3.

D.2.1 Limiter l’introduction de connaissances *a priori*

La limitation de l’introduction de connaissances *a priori* doit permettre d’éviter d’avoir un système qui recopie les raisonnements du clinicien en disposant des mêmes connaissances. Si nous arrivons à ce résultat nous échouons dans l’objectif de mettre en collaboration des systèmes qui ont des points de vue différents.

De plus, l’introduction d’une connaissance *a priori* est un ajout d’une contrainte dans un espace de recherche. Par conséquent, l’introduction de connaissances *a priori* est une limite à la créativité du système. En restreignant trop un espace de recherche on limite l’émergence de nouvelles connaissances. Au pire, on ne fait que redécouvrir ce qui a été introduit au travers des contraintes définies *a priori*.

Cependant, l’introduction de connaissances *a priori* est nécessaire pour que ce qui est construit par le système puisse permettre au système de converger. Les interactions (indirectes) avec le clinicien permettent de conduire le processus émergent à produire des connaissances qui soient compatibles (*i.e.* cohérentes) avec celles du clinicien. Ainsi, les annotations réalisées par le clinicien doivent être intégrées aux autres annotations que le système réalise. Le processus émergent met alors en cohérence à la fois ses connaissances, mais également ses connaissances et les annotations du clinicien.

Les phénomènes émergents tiraient leur capacité à générer de nouvelles formes complexes par un bon équilibre entre la dynamique des interactions entre entités et les contraintes imposées par le niveau supérieur. Pour l'émergence d'une interprétation propre, il faut également une introduction équilibrée de connaissance *a priori*.

D.2.2 Adapter la “perception” et la “cognition” de la machine

L'adaptation des modèles informatiques qui servent de support aux traitements des données par le système permet l'adaptation d'une forme de “perception” et de “cognition”.

Une machine informatique est un calculateur. Par conséquent, si on peut parler de “cognition” à propos d'une machine, il s'agit des *traitements* algorithmiques qui peuvent être réalisés. Sur le schéma de la figure 2.4 page 50, les traitements qui peuvent être réalisés par le système sont représentés par les flèches entre les niveaux d'abstraction. Ces traitements utilisent à la fois des algorithmes et des modèles (d'évènements et de scénarios). L'adaptation des traitements peut donc passer par l'adaptation des algorithmes (les algorithmes eux-mêmes ou leurs paramètres) ou les modèles qui y sont utilisés.

Les capacités de perception sont caractérisées par les affordances qui permettent à un système de voir le monde dans lequel il est situé. Pour notre système, la “perception” implique de rendre les données traitables, *i.e.* rendre le monde accessible à la cognition. Elle est donc également liée aux modèles qui permettent à la machine de s'appropriier les données et les annotations.

Dans notre système, les algorithmes sont définis et non modifiables. Les modèles sont alors les éléments qui permettent d'adapter à la fois la “perception” des données et leur “cognition” par le système. Le système conçu pour collaborer sur le modèle d'un couplage structurel devra donc être capable d'adapter ses modèles.

D.3 Collaboration dans l'interprétation de séries temporelles

Nous revenons à présent à notre problématique d'interprétation collaborative de séries temporelles. Pour cela, nous avons à concrétiser nos réflexions sur la conception de l'interprétation de séries temporelles avec celles de la collaboration inspirée du paradigme de l'Enaction. Nous proposons donc ici notre conception de la collaboration pour l'interprétation de séries temporelles par une machine et un clinicien. Mais d'abord, nous revenons plus particulièrement sur les annotations qui peuvent être réalisées sur des séries temporelles pour permettre l'échange d'informations.

D.3.1 Sémantique annotationnelle

Nous formalisons maintenant les annotations qui peuvent être échangées par la machine et le système en particulier pour mettre en évidence le sens qui peut être porté par les différentes annotations, dans les termes de Bachimont [16], il s'agit de mettre en place une sémantique annotationnelle.

Dans notre cas, une annotation n'a pas de contenu textuel, il s'agit principalement d'une ancre ayant un caractère symbolique attaché à une partie d'un enregistrement (une partie d'un signal ou de plusieurs signaux en parallèle). Le contenu textuel n'est pas facilement utilisable par la machine et ne fait donc pas partie des éléments qui peuvent servir à la communication entre les agents. Le sens intrinsèque qui peut être porté par une annotation et le lien qu'elle crée avec un modèle (d'évènement ou de scénario) permet de transmettre une information entre les agents. En procédant ainsi, les annotations elles-mêmes sont constamment réinterprétées par les agents et donc réappropriées.

Nous utilisons les éléments de l'interprétation (segments, symboles et scènes) comme marques pouvant être utilisées par le clinicien ou par le système pour annoter les séries temporelles. Les segments ne sont pas porteurs de sens, mais permettent de focaliser l'attention sur des parties intéressantes d'un signal.

Les symboles ont pour cible une partie d'un signal. Un symbole est associé à un modèle d'évènement ce qui permet une transmission des informations sur les modèles d'évènement par son intermédiaire. D'une part, le modèle d'évènement apporte le sens à l'annotation sous forme de symbole. L'échange de symboles permet donc d'échanger à propos d'interprétation d'une série temporelle. Mais d'autre part, une annotation de symbole donne un représentant du modèle d'évènement auquel il est associé. L'échange de symboles permet donc d'échanger à propos des modèles : par similarités et différenciations des annotations

de symboles, le clinicien et le système peuvent ainsi communiquer sur les événements et leurs modèles sans avoir à échanger directement ceux-ci.

Finalement, les scènes réalisent l’ancrage des scénarios (de même que le font les symboles avec les événements) mais la cible est cette fois une portion d’un enregistrement de patient.

L’utilisation d’annotations sous des formes *symboliques* comme le sont les segments, les symboles et les scènes peut apparaître en opposition avec la conception non-symbolique de la pensée dans l’approche cognitive de Varela. La situation est en fait la même que pour l’exemple pris par Dionisi et *al.* [81] d’un jeu d’échec joué par un homme et une machine : “Pour le logiciel, les notions d’échiquier, de pièce, de déplacement, n’ont aucun sens. Ces symboles sont destinés à la pratique cognitive. La démarche de caractérisation d’un logiciel de jeu d’échec destiné à entrer en collaboration avec les processus cognitifs d’un joueur doit tenir compte de ce que peut être un tel problème vu par le logiciel lui-même. Et notamment, considérer quelles doivent être les conditions à mettre en place pour que le processus logiciel soit apte à atteindre le niveau d’abstraction convenable, échappant à la logique d’un raisonnement humain, mais parfaitement adapté à la logique machine. L’essence du jeu d’échec est totalement abstraite et conceptuelle, et se réifie différemment en direction des deux processus qui y accèdent, le cognitif et le logiciel.”

De la même manière, il nous a fallu opérationnaliser les éléments constitutifs d’une interprétation afin de les rendre à la fois maîtrisables par les utilisateurs et par le système. Pour le système, la représentation symbolique peut être appréhendée au travers du schéma d’interprétation présenté en section B.2. Pour le clinicien ce type d’annotation semble correspondre à une pratique, mais c’est une réduction de la sémantique riche qu’il pourrait utiliser.

D.3.2 Schéma de l’interprétation collaborative de séries temporelles

Le schéma général de l’interprétation collaborative de séries temporelles donné dans la figure 2.11 est le résultat de nos réflexions sur la notion d’interprétation et sur la collaboration Homme/Machine. L’étude de l’interprétation nous a amené à proposer le schéma global de la figure 2.4 page 50 pour réaliser une interprétation de séries temporelles. L’étude de la collaboration Homme/Machine nous a amené à concevoir la collaboration comme deux couplages structurels avec l’environnement, illustré par le schéma de la figure 2.9 page 61.

Dans le schéma de la figure 2.11, les données et les annotations constituent l’environnement à travers lequel le système et l’utilisateur échangent des annotations. Chaque agent produit de manière autonome son interprétation des séries temporelles et échange au travers d’un environnement commun constitué de séries temporelles et des abstractions construites par les annotations des séries : séries segmentées, séries symboliques et scènes.

Les modèles d’événements et les scénarios sont construits à partir des données indépendamment par chacun des agents. Sur la figure, nous représentons les modèles du clinicien de manière symétrique à ceux que nous utilisons pour la machine mais nous avons vu que la connaissance du clinicien ne prend pas une telle forme représentationnelle du monde. La figure vise principalement à mettre en évidence que les deux agents participent à chaque niveau à la construction de l’interprétation.

Pour mettre en place effectivement cette collaboration, il nous faut maintenant concevoir le système qui va permettre de réaliser le schéma d’interprétation présenté en respectant les contraintes fixées dans la section D.2.

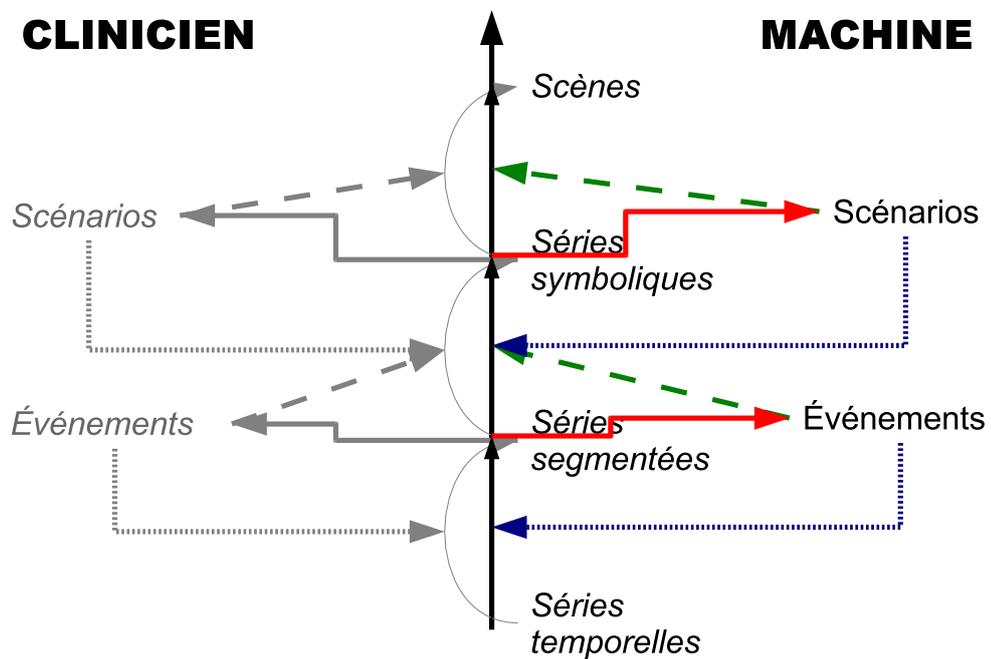


FIG. 2.11.

Interprétation collaborative de séries temporelles. Les deux agents (clinicien et machine) réalisent chacun indépendamment une interprétation des séries temporelles au moyen d'événements et de scénarios. Ils collaborent au travers de l'environnement partagé que constituent les données et leurs annotations. Pour la machine nous mettons en évidence les processus de construction de l'interprétation que nous allons utiliser, pour le clinicien nous les grisons car il ne s'agit que d'une modélisation de son raisonnement.

Chapitre 3

Un système de support à l'interprétation collaborative de séries temporelles

On présente maintenant le système pour l'interprétation collaborative de séries temporelles. Les réflexions menées au chapitre 2 nous ont conduits à décrire des contraintes de conception du système de sorte à pouvoir envisager qu'il puisse se coupler avec l'utilisateur pour collaborer à l'interprétation efficacement.

On utilise le paradigme des systèmes multi-agents pour concevoir notre système. Ce paradigme permet d'une part de mettre en place une organisation des traitements en une hiérarchie de triades et d'agents qui permet l'émergence d'une interprétation autonome, et d'autre part d'intégrer facilement les interactions de l'utilisateur.

Ensuite, ayant pointé dans les chapitres précédents l'importance des modèles, on présente les formalismes dans lesquels sont écrits les modèles d'évènements et de scénarios. Ces formalismes peuvent être construits par le système et être utilisés pour l'annotation des séries temporelles.

Enfin, on présente les solutions algorithmiques qui sont implémentées dans les agents.

Rappelons brièvement l'objectif du système que nous présentons. Nous cherchons à développer un système qui va permettre de collaborer avec un clinicien pour interpréter des séries temporelles. L'interprétation et la collaboration telles que nous les avons conclues dans le chapitre 2 nous ont amenés à définir des contraintes sur le système informatique.

Le système doit être capable de construire une interprétation des séries temporelles de manière autonome à partir des séries temporelles et des annotations (segments, symboles et scènes) à disposition mais également être capable de produire des annotations pendant son interprétation et intégrer de nouvelles informations.

Le système doit être fortement adaptable. D'une part, il doit être capable d'intégrer dynamiquement de nouvelles informations au cours de son interprétation mais également être capable d'adapter ses traitements (*i.e.* ses modèles) pour focaliser ses capacités de perception des données et des annotations.

Finalement, le système doit être capable d'échanger des annotations avec le clinicien. Son adaptation aux nouvelles annotations (ou à des suppressions) constitue son moyen de recevoir de l'information de la part du clinicien. En outre, le système doit être capable de produire automatiquement ces propres annotations.

Le système que nous présentons ici est une proposition pour remplir ces trois pré-requis à la conception de notre collaboration. Dans la première section, nous présentons l'architecture générale qui est un système multi-agents (SMA) afin d'illustrer le fonctionnement global du système seul et avec un utilisateur. Dans la section suivante nous détaillons les modèles qui sont utilisés. Nous y consacrons une section entière car nous avons déjà pu indiquer l'importance du choix des modèles dans notre entreprise de *découverte* d'une interprétation. Finalement, nous donnons des détails techniques sur les algorithmes qui sont utilisés dans nos agents pour traiter les données.

A Architecture générale : Un système multi-agents

L'objectif de l'utilisation des systèmes multi-agents est de faire *émerger* l'interprétation. Comme on a pu le présenter dans la Chapitre 2 section A.2, en plus de la structure hiérarchique et l'utilisation d'une rétro-action (*feedback*) entre les niveaux d'abstraction telle que nous le proposons dans la construction de l'interprétation, nous avons besoin d'avoir une dynamique d'interactions "complexes" pour qualifier le processus d'émergent. Les systèmes multi-agents permettent une distribution des traitements dans des entités **autonomes** que sont les agents. En particulier, il n'y a pas de contrôle centralisé des traitements des différents agents.

Nous présentons tout d'abord une description de notre système multi-agents selon la méthodologie "Voyelle" (AEIO) de Demazeau [163]. Cette méthodologie invite à décrire le SMA par les quatre grandes dimensions qui le composent :

- A (Agents) : Quel(s) type(s) d'agents sont utilisés? Quel(s) sont leur(s) compétences et buts propres?
- E (Environnement) : Quels sont les caractéristiques de l'Environnement partagé dans lequel évoluent les agents,
- I (Interactions entre les agents) : Comment les agents communiquent entre eux? Qu'est-ce qui est échangé?
- O (Organisation) : Comment les agents sont-ils structurés pour accomplir la tâche globale du système?

Ensuite, nous ajoutons le "U" de la méthode en présentant les interactions du SMA avec l'Utilisateur dans le cadre de l'interprétation collaborative présentée dans le Chapitre 2.

A.1 Description du SMA

Notre système étant centré à la fois sur l'apparition **émergente** d'une interprétation, et sur la construction de connaissances **cohérentes**, le point central de notre système est son organisation. C'est de l'organisation des traitements interdépendants que nous allons pouvoir faire réaliser une *interprétation* globale par le SMA.

Donc nous commençons par une description brève des agents obtenus à partir de l'agentification du processus d'interprétation. Ensuite, pour décrire notre système selon la méthode AEIO.

A.1.1 Agentification du processus d'interprétation

Dans le chapitre 2, nous avons présenté le schéma global 2.4 de l'interprétation d'une série temporelle sur les quatre niveaux d'abstractions (le schéma est partiellement reproduit à la page suivante) :

- Séries temporelles : ce sont les données brutes enregistrées sur nos patients. Un enregistrement de patient comporte plusieurs séries temporelles en parallèle. Un enregistrement est donc une série multivariée ;
- Séries segmentées : les segments identifient les zones d'intérêt dans les séries temporelles. Un segment ne précise pas en quoi cette partie est intéressante. Les séries segmentées sont obtenues par la **segmentation** des séries temporelles ;
- Séries symbolisées : à ce niveau, les zones d'intérêts sont mises en correspondances avec des événements. Chaque événement pouvant être associé à une valeur symbolique (*e.g.* son nom ou une référence dans une ontologie), la concaténation de ces symboles datés dans la série constitue une série symbolique. Cette étape est la **symbolisation** des séries temporelles ;
- Scènes : Les scènes sont des instances de scénarios qui ont pu être identifiées dans un enregistrement (*i.e.* transversalement à toutes les séries temporelles d'une série multivariée). Par conséquent, les séries temporelles peuvent être résumées de manière très abstraite par les scènes qui auront été reconnues dans les séries temporelles d'un enregistrement. La **reconnaissance des scénarios** dans les séries symboliques permet de construire ces scènes.

La segmentation, la symbolisation puis la reconnaissance de scénarios constituent le chemin élémentaire pour construire les niveaux de plus en plus abstraits. Ils utilisent pour cela les modèles d'événements et les scénarios (flèches en tirets sur la figure 2.4). Mais le système devant être capable de réaliser l'interprétation des séries temporelles de façon autonome, il doit également construire ces modèles à partir des données (flèches pleines sur la figure 2.4) : les modèles d'événements doivent être construits à partir des segments d'intérêt identifiés dans les séries segmentées (par des méthodes de classification non-supervisées des segments) et les scénarios doivent être identifiés à partir des séries symboliques (par une méthode d'apprentissage non-supervisée de scénarios).

Enfin, la machine construisant une interprétation cohérente des données, elle doit s'assurer que les modèles qui sont construits et les données sont globalement cohérents. Un mécanisme de *feedback* entre les niveaux (flèches pointillées sur la figure 2.4) est donc utilisé pour progressivement améliorer les abstractions des séries temporelles à la manière de Heutte et *al.* qui utilisent un système multi-agents intégrant du *feedback* pour l'amélioration des performances d'un système de reconnaissance de caractères [112].

Nous proposons ainsi un système dans lequel il y a six types d'agents chaque type d'agents doit réaliser l'une des tâches nécessaires à l'interprétation autonome des séries temporelles :

- agents de segmentation,
- agents de construction de modèles d'événements,
- agent de symbolisation (depuis les événements),
- agents de constructions de scénarios,
- agents de reconnaissance de scénarios (et de correction de la symbolisation depuis les scénarios),
- agent de scène.

Ces agents sont le résultat d'une agentification des traitements illustrée par la figure 3.1. Les différents traitements sont répartis par différents agents qui implémenteront les méthodes correspondantes.

L'utilisation de la méthode AEIO va permettre de passer de la définition abstraite de ces agents à la conception du système dans sa globalité. Nous commençons par présenter son organisation. Puis, nous reviendrons sur une description plus fine des agents à la lueur de cette organisation. Et finalement, nous décrirons l'environnement des agents et les interactions qui se font via l'environnement.

A.1.2 Organisation : Hiérarchie de triades

Une première solution pourrait consister à construire un système multi-agents dans lequel tous les agents cohabitent et interagissent entre eux (solution sans organisation, ou bien qui peut être qualifiée d'auto-organisée). Charge à chacun d'identifier l'agent qui sera capable de lui fournir les informations qui lui sont nécessaires. Mais dans une telle solution, on peut craindre que les agents perdent plus de temps à chercher une information qu'à la traiter.

Nous avons retenu une solution qui consiste à utiliser l'opérationnalisation de l'interprétation de séries temporelles pour organiser les agents. Nous reprenons donc dans la figure 3.2 une organisation du système

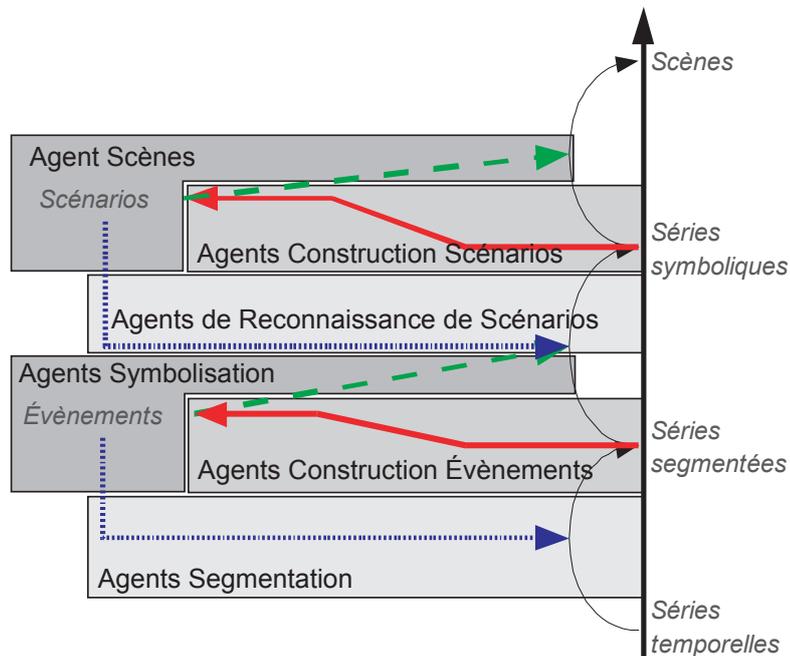


FIG. 3.1. Agentification des traitements des données par niveaux d'abstraction. Chaque agent est en charge d'un des niveaux de traitement des séries temporelles.

multi-agents par une hiérarchie de triades directement inspirée de la figure 2.6 page 51.

Dans cette vue conceptuelle, le système est organisé comme une hiérarchie de triades. Le système est globalement une triade constituée de deux entités de niveau inférieur (Traduction symbolique et Scènes) qui sont elles-mêmes des triades. La triade Traduction symbolique est constituée des triades de Segmentation et de Construction de modèles. La triade de Scènes est constituée des triades de Reconnaissance de scénarios et de Construction de scénarios.

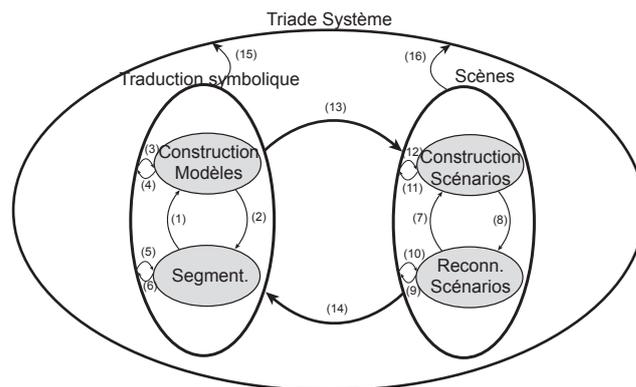


FIG. 3.2. Vision conceptuelle de l'architecture générale du système : Une hiérarchie de triades. Les liens numérotés indiquant les échanges d'informations sont décrits dans le tableau 3.1 page 78.

Chaque triade représente une tâche à accomplir, mais les traitements ne sont pas réalisés par les triades elles-mêmes, mais par les agents qu'elles contiennent.

Une triade ne contient que des agents d'un même type correspondant à la tâche qu'elle réalise. Ainsi

tous les *agents de segmentation* sont dans la triade de Segmentation, tous les *agents de Reconnaissance de scénarios* sont dans la triade Reconnaissance de scénarios, etc. Les triades grisées contiennent plusieurs agents tandis que les triades englobant d'autres triades (Triades Scènes, Traduction Symbolique et Système) ne contiennent qu'un unique agent qui réalise les traitements. Il s'agit d'un choix de conception *a priori* lié au fait qu'initialement ces traitements étaient réalisés par la triade elle-même et qu'à la suite de l'agentification de tous les traitements, un unique agent était nécessaire pour réaliser la tâche des triades intermédiaires. Mais on pourrait très bien avoir plusieurs agents pour réaliser la symbolisation des séries temporelles.

Une triade n'intervient que pour orchestrer les traitements faits par les agents :

- Dans le cas d'une triade ne contenant que des agents : les agents sont successivement exécutés. En pratique, pour les triades ne contenant que des agents, un tirage aléatoire permet de ne pas exécuter les agents dans le même ordre. Si des agents sont à créer ou supprimer, ils le sont au début d'un cycle.
- Dans le cas d'une triade intermédiaire, l'agent et les deux triades inférieures sont successivement exécutés. L'ordre d'exécution des trois entités est toujours le même (l'agent est exécuté en dernier pour répercuter immédiatement les nouvelles modifications amenées par les traitements).

Lorsqu'un agent est exécuté, il réalise une unique fois le traitement spécifique à cet agent en suivant un schéma commun à tous les agents (*cf.* la définition de la dynamique des agents dans la section A.2.1). On peut finalement représenter les traitements réalisés au cours d'un cycle du système comme un arbre de récursion représenté par la figure 3.3. Ce cycle global est alors être exécuté en boucle. En faisant réalisé en boucle ce cycle, les agents (*i.e.* les traitements) sont également exécutés en boucle.

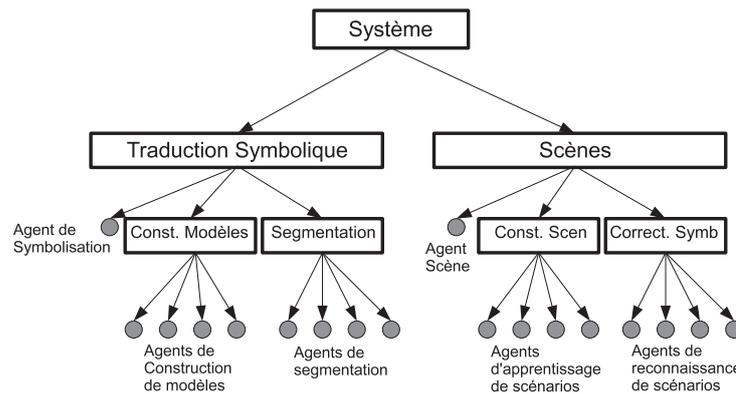


FIG. 3.3.

Représentation des relations de dépendance entre les processus que sont les agents et les triades. La structure des dépendances est hiérarchique, directement inspirée de la hiérarchie en Triade de la figure 3.2.

A.1.3 Définition des agents

Les agents sont ici des unités de traitement élémentaires : un agent est consommateur d'une information et producteur d'une autre. La conception des agents ne se rapporte pas à des modèles d'agents particuliers (*e.g.* agents BDI, ...) ou de patrons (*e.g.* ADELFE [25], ...). Il s'agit d'une conception *ad hoc* des agents que nous décrivons dans la section A.2.

Ces unités prennent le titre d'agent dans la mesure où chacune de ces entités est autonome et dotée de compétences propres, qu'elle a la capacité d'interagir avec d'autres agents (*cf.* section A.1.5) et qu'elle perçoit et agit dans son environnement (*cf.* section A.1.4).

Les agents se distinguent par les traitements qu'ils ont à réaliser. Pour chaque type d'agent, on présente dans les paragraphes suivants leurs buts, les modèles qui leurs sont nécessaires pour réaliser leur tâche. Les méthodes qui ont été mises en œuvre pour que les agents réalisent effectivement ces spécifications sont données dans la section C.

Agents de segmentation Un *agent de segmentation* doit segmenter les séries temporelles, *i.e.* trouver des segments d'intérêt dans une série temporelle. Un *agent de segmentation* n'a pas pour but de construire

une partition d'une série temporelle.

On distingue deux types d'*agents de segmentation* : ceux qui sont fondés sur des modèles déjà existants et ceux qui cherchent de nouveaux segments sur des critères particuliers ne nécessitant pas de connaissances *a priori* fortes (e.g. méthodes statistiques). Ces agents doivent permettre initialement d'annoter des segments intéressants et par la suite d'amener de la nouveauté dans les segments déjà recherchés. Les premiers sont créés dynamiquement par le système et les seconds sont permanents.

Un *agent de segmentation* du premier type est construit par l'*agent de symbolisation* (en tant que triade mère) pour chaque modèle d'évènement disponible. Les modèles sont le produit des *agents de construction de modèles*. L'*agent de segmentation* embarque alors ce modèle. Le but de l'agent est donc de retrouver dans les séries temporelles les zones qui pourraient correspondre à ce modèle. De telles zones sont alors annotées comme un segment potentiellement d'intérêt pour la suite du traitement. (cf. section C.1 pour plus de détails)

Agents de construction de modèles Il y a un agent pour la construction de modèles d'évènements par type de séries qui composent une série multivariée. Ces agents sont donc créés initialement et il n'y a pas d'insertion ou de suppression dynamique de ces agents. L'agent (1) récupère l'ensemble des segments qui ont été trouvés par celui-ci dans son environnement, (2) construit des classes de segments homogènes et (3) extrait des modèles à partir des traits caractéristiques de chaque classe. (cf. section C.2 pour plus de détails)

Agent de symbolisation Le but de l'*agent de symbolisation* est de traduire les séries temporelles en séries symboliques, c'est-à-dire d'annoter les séries temporelles à l'aide de symboles. Le seul agent de symbolisation traite tous les enregistrements. Il est également en charge d'apporter des modifications aux séries symboliques à partir des interactions avec la triade des Scènes. Des suppressions ou des ajouts de ces agents sont réalisés par le système lorsque l'utilisateur supprime ou ajoute des enregistrements à traiter.

Agents de construction de scénarios Un *agent de construction de scénarios* (ou d'apprentissage de scénarios) doit construire des scénarios à partir des séries symboliques. Chaque *agent de construction de scénario* est spécifiquement associé à un évènement. L'agent a alors la charge de construire une *explication* des occurrences de l'évènement sous la forme de scénario (cf. section B.4.2). Des *agents de construction de scénarios* sont ajoutés et supprimés au cours du traitement des données en fonction des évènements qui ont été identifiés jusque là. (cf. section C.3 pour plus de détails)

Agents de reconnaissance de scénarios Les *agents de reconnaissance de scénarios* parcourent les séries symboliques à la recherche d'occurrences d'un des scénarios trouvés par les *agents de construction de scénarios* (Pour les mêmes raisons que celles qui nous ont conduit à séparer la symbolisation des autres traitements, la reconnaissance est réalisée indépendamment par les *agents de reconnaissance de scénarios*. (cf. section C.4 pour plus de détails)

Par la réalisation de cette reconnaissance de scénarios, l'agent permet d'une part de construire des annotations de scènes sur les enregistrements, mais également d'apporter des modifications des séries symboliques qui vont améliorer la confiance¹ dans les scénarios appris par le système. Lorsqu'il réalise une reconnaissance partielle d'un scénario, il peut proposer des modifications de sorte à améliorer la confiance dans leur scénarios. Il procède ainsi à un raffinement (*feedback*) des séries symboliques qui seront de nouveau utilisées pour un apprentissage.

Agent de Scènes L'*agent Scènes* a un rôle d'intermédiaire. Il regroupe les annotations de scènes pour les valider et transmet les modifications des séries symboliques réalisées à la triade de Symbolisation pour qu'elle en prenne compte dans la suite de ses traitements.

Agent Système A ce niveau, un tel agent aurait pour but de regrouper les différentes annotations. En pratique, nous n'utilisons pas d'*agent Système*.

¹ Il s'agit d'une mesure de confiance calculée par la fréquence du scénario obtenue dans l'algorithme d'apprentissage, cf. section C.4.

La dynamique de création, suppression et modification des agents pendant l'interprétation est donnée dans la section A.2.

A.1.4 Définition d'un environnement

L'environnement est composé des données et des connaissances à traiter par les agents, c'est-à-dire des séries temporelles et de leurs annotations. L'environnement est commun à tous les agents. On peut définir une "topologie" à trois dimensions de cet environnement :

1. la position dans le temps d'une série temporelle,
2. la série dans une série multivariée traitée,
3. l'enregistrement (série multivariée) traité.

Cette topologie permet d'accéder aux annotations qui sont déposées dans l'environnement, et qui du coup en font partie.

Les modèles ne font pas partie de l'environnement, ils sont conservés dans des agents. Un modèle d'évènement est intégré dans un *agent de segmentation* et est également conservé par l'*agent de traduction symbolique*. Un modèle de scénario est intégré dans un *agent de reconnaissance de scénario* et est également conservé par l'*agent des Scènes*. Nous reviendrons ensuite sur la raison de ne pas mettre ces modèles dans l'environnement.

Ainsi défini, l'environnement 1) permet de situer à un instant donné un agent dans celui-ci et 2) peut évoluer par les traitements des agents. Ces derniers peuvent percevoir et agir dans cet environnement.

Dans notre cas, les agents sont situés par rapport aux données qu'ils traitent. Tous les agents n'ont alors qu'une vue partielle à un moment donné de l'environnement, c'est ce qui permet d'indiquer qu'ils sont localisés dans celui-ci. Par exemple, un *agent de segmentation* qui parcourt une série temporelle est localisé à un endroit de cette série. Cette position s'accompagne d'un champ de perception limité. Sur ce même exemple, l'agent ne peut "voir" qu'une partie de la série et n'aura jamais connaissance des annotations autres que celles qu'il produit (segments des autres *agents de segmentation*, symboles ou scénarios) et même de toutes les séries qui ne lui sont pas attribuées à traiter (d'un type différent).

Les agents peuvent également agir sur l'environnement au moyen des traitements puisqu'ils peuvent réaliser des annotations. Donc en réalisant une annotation, ils modifient l'environnement, ce qui est le propre d'une action. Des travaux similaires sur la construction d'environnements *conceptuels* ont déjà été proposés par S. Hassas et *al.* [9, 34], qui transposent les approches écologistes [102] à la résolution de problèmes par émergence.

A.1.5 "Interactions" : Flux d'informations entre les agents

Il y a deux types d'informations qui peuvent être échangées par les agents : les annotations et les modèles. Il en découle deux types d'interactions :

- les annotations sont échangées au travers de l'environnement. Lorsque les agents produisent une annotation (ou une modification d'une annotation), elle est disposée dans l'environnement. L'environnement étant commun à tous les agents, ceux qui sont concernés peuvent alors y accéder.
- les modèles sont directement échangés par des interactions directes entre agents.

Lorsque les agents échangent des modèles, il y a deux situations particulière qui sont réalisés uniquement par l'*agent de Traduction Symbolique* et par l'*agent de Scènes*. Soit ces agents veulent récupérer un modèle auprès d'un des agents qui les a construit (dans une triade inférieur) et dans ce cas il va directement dans la "mémoire" de l'agent pour la récupérer. Soit ils créés un agent (de *segmentation* ou de *reconnaissance de scénarios*) à partir d'un modèle qu'il possède.

Dans notre système, il n'y a pas d'échange d'information entre agents de même nature. De plus, les échanges d'informations sont contraints par l'organisation générale du système. Ils se limitent donc aux seuls liens figurés sur la figure 3.2. Le tableau 3.1 résume les flux d'information entre les différents types d'agents. Les chiffres indiqués pour les différentes informations qui peuvent être échangées correspondent aux numéros des liens indiqués sur la figure 3.2.

Agent	Depuis les agents des triades inférieures	Des agents de la triade opposée	À/De des agents de la triade supérieure
Segmentation		(1) Segments	(6) Séries segmentées / (5) Modification de la segmentation
Construction de modèles d'évènements		(2) Modèles d'évènements	(4) Ensembles de modèles d'évènements / (3) Modifications de la traduction symbolique de segments
Symbolisation	(6) Séries segmentées et (4) ensemble de modèles d'évènements	(14) Séries symboliques raffinées	(15) et (13) : Séries symboliques /
Construction de scénarios		(7) Séries symboliques raffinées	(11) Scénarios / (12) Modifications des scénarios
Reconnaissance de scénarios		(8) Scénarios	(9) Scènes et Modifications des séries symboliques / (10) Modification des scènes
Scènes		(13) Séries symboliques	(16) Scènes /
Système	(15) Séries symboliques et (16) Scènes		Interprétation /

TAB. 3.1.

Résumé des flux d'information entre les agents. Les chiffres entre parenthèses font références aux flèches de la figure 3.2.

A.2 Dynamique des agents

Le système proposé est animé par la dynamique des agents individuellement mais également par la dynamique globale de la construction de l'interprétation qui conduit le système à créer et supprimer des agents. La description de la dynamique permet de mieux comprendre le fonctionnement global du système pour la construction d'une interprétation cohérente. La construction de l'interprétation cohérente impose que la dynamique du système assure une transmission des informations entre les niveaux comme le proposait la figure 3.1 et que le système réduise les inconsistances entre les éléments atomiques de l'interprétation alors qu'ils sont créés indépendamment par des agents différents. D'autre part, nous devons mettre en évidence (à défaut de preuve formelle) que le système permet de converger en temps fini vers une interprétation.

On décrit tout d'abord le fonctionnement interne d'un agent qui a été conçu *ad hoc* pour nos besoins d'agents simples. Ensuite, nous décrivons la dynamique de création et de suppression des agents au cours de l'interprétation pour argumenter quant à la construction d'une interprétation cohérente.

A.2.1 Schéma de fonctionnement d'un agent

Le schéma du fonctionnement d'un agent est le suivant :

1. Récupérer localement dans l'environnement les informations nécessaires pour le traitement et les données à traiter ;
2. Traiter les données ;
3. Mettre à disposition des connaissances obtenues (dans l'environnement, ou dans sa "mémoire partagée" pour les modèles) ;

Chaque agent à un comportement très autonome et un agent ne prévient personne d'une mise à jour dans l'environnement des informations (plus récente) qu'il produit. Chaque agent qui a besoin de cette information récupérera cette information modifiée lors de son prochain traitement.

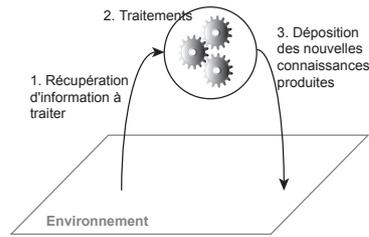


FIG. 3.4. Illustration du fonctionnement d'un agent.

A.2.2 Dynamique de création et suppression des agents

On illustre la dynamique de création et suppression des agents par un exemple du fonctionnement du système à son initialisation. Dans cet exemple, on suppose que les enregistrements sont constitués de deux séries temporelles. L'exemple est donné par la figure 3.5 et fait référence à la figure 3.3 pour représenter l'apparition des agents dans l'arbre des triades.

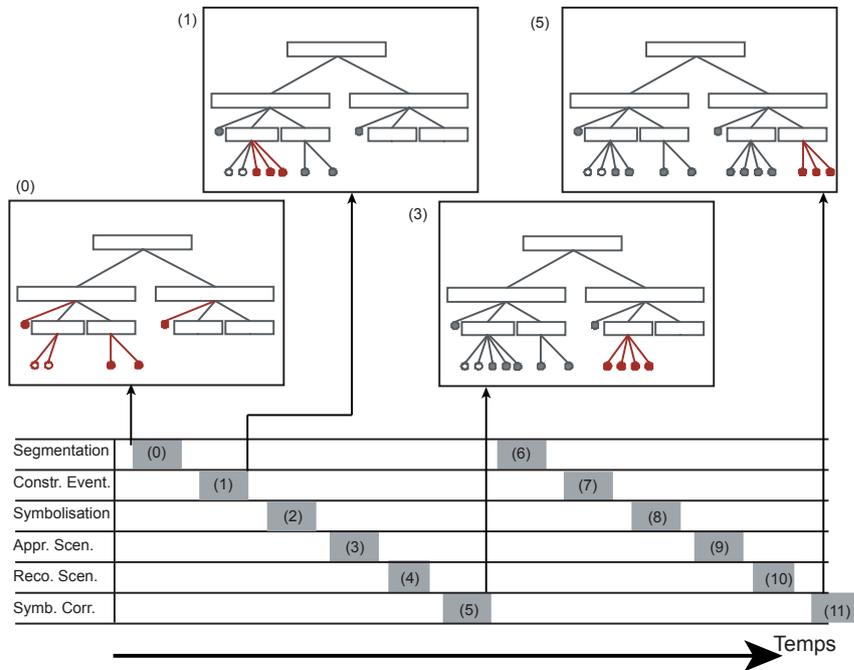


FIG. 3.5. Illustration du fonctionnement de la dynamique de création et modification des agents dans le MAS. En bas de la figure, un exemple de succession de traitements par les triades est donné sous la forme de chronogramme. En haut de la figure, on illustre l'apparition des agents par l'apparition de nouveaux *points* dans la hiérarchie de triades (cf. figure 3.3).

L'étape (0) illustre la scène initiale du SMA. Deux *agents de construction de modèles* sont présents, un par type de séries temporelles. Des *agents de segmentation* (clairs) sont des *agents de segmentation* permanents qui réalisent les segmentations sur critères statistiques. Enfin, les triades intermédiaires disposent de leur propre agent pour réaliser les traitements spécifiques à ces triades. Tous ces agents sont initialement introduits dans le SMA et ne sont jamais supprimés ou ajoutés.

Initialement, le système n'a à disposition que des séries temporelles, et les seuls agents qui peuvent travailler sont les *agents de segmentation* (sans modèles) qui produisent des segments. À l'étape (1), c'est au tour des *agents de construction de modèles* de rentrer en action pour créer des modèles d'événements. De ces modèles sont construits des *agents de segmentation* qui seront en charge par la suite de retrouver ces modèles dans les séries temporelles. À l'étape (1), on voit donc apparaître de nouveaux *agents de segmentation* dans le SMA (sur la figure 3.5, on figure l'apparition de nouveaux agents comme de nouveaux

points dans la hiérarchie illustrée dans les encadrées). Ces nouveaux agents sont mis à contribution pour produire de nouveaux segments lors du prochain cycle (Étape (6)). À l'étape (2), la symbolisation est lancée par la triade de Symbolisation : à partir des nouveaux segments et des modèles d'évènements, l'*agent de symbolisation* construit des séries symboliques. Ces séries symboliques sont ensuite déposées dans l'environnement pour la triade des Scènes. Pour les étapes (3) et (4), il n'y a pas d'agent, et donc pas de traitement à faire. En effet, les *agent de construction de scénarios* sont créés à l'étape (5) : l'*agent scènes* identifie les nouveaux types d'évènements qui sont apparus dans l'environnement et de nouveaux *agents de construction de scénarios* sont créés. Chaque *agent de construction de scénarios* explique un évènement qui lui est spécifique. Ensuite, les étapes (6) à (8) reprennent les mêmes traitements que précédemment (il s'agit d'un second cycle). Le résultat de cette nouvelle construction va :

- faire apparaître de nouveaux modèles, alors de nouveaux *agents de segmentation* sont introduits,
- supprimer d'anciens modèles, les *agents de segmentation* associés à ce modèle sont alors détruits,
- faire évoluer les autres modèles, les *agents de segmentation* correspondant récupèrent alors de nouveaux modèles.

À l'étape (9), les *agent de construction de scénarios* existent et donc des scénarios sont construits. Mais il y a toujours aucun scénario à reconnaître à l'étape (10). De nouveaux scénarios ont été construits en (9) et donc de nouveaux *agents de reconnaissance de scénarios* sont créés seulement à l'étape (11) pour les rechercher dans les séries symboliques.

Par la suite, en fonction des nouveaux évènements qui apparaissent et disparaissent, ces *agents de construction de scénarios* sont créés ou sont supprimés. Les *agents de reconnaissance de ces scénarios* sont également détruits lors de la suppression des évènements qu'ils cherchent à expliquer, mais ils sont créés lorsqu'une explication de l'évènement est construite par les *agents de construction de scénarios*.

Au fur et à mesure de la construction d'une interprétation qui gagne en cohérence, certains agents vont réduire leur nombre de modifications de l'environnement car ils n'ont pas à remettre en cause les traitements précédents. Par conséquent, les perturbations qu'ils vont engendrer auprès d'autres agents seront également moins importantes. L'interprétation sera globalement cohérente lorsqu'aucun agent n'a plus de mise à jour à faire des connaissances qu'il peut produire.

A.3 Interactions avec l'utilisateur

Nous décrivons à présent les interactions du SMA avec l'utilisateur. Nous précisons tout d'abord comment vont se positionner ces interactions dans notre structure hiérarchique en triades et ensuite nous détaillons les différentes interactions qui peuvent être utilisées par le clinicien et le système pour échanger des informations.

A.3.1 Utilisateur situé dans la hiérarchie

Le système et la machine forment un ensemble pour lequel la capacité à mener une interprétation est émergente. L'organisation de la collaboration est donc également triadique et les interactions entre le système et l'utilisateur constituent un nouveau niveau triadique dans la hiérarchie de la figure 3.2.

Dans la mesure où l'utilisateur agit directement à chaque niveau de traitement des données, on ne peut pas le représenter comme une relation entre le système dans sa globalité. Les interactions entre le système et l'utilisateur se positionnent dans la hiérarchie de triade comme le niveau le plus inférieur (*cf.* figure 3.6). L'interaction est ainsi située dans chaque tâche qui permet la construction de l'interprétation des séries temporelles.

Il y a donc des interactions qui permettent de réaliser les tâches :

- de segmentation,
- de construction des classes d'évènements,
- de symbolisation (construction des séries symboliques),
- de construction des scénarios,
- et de reconnaissance des scénarios.

Les interactions à chaque niveau peuvent alors être décrites indépendamment les unes des autres. Mais pour l'utilisateur, il aura en permanence accès aux interactions de tous les niveaux. De même pour les systèmes de parcours de séries temporelles présentés dans la section D chapitre 1, les interactions peuvent se faire au niveau de la segmentation, de la symbolisation ou des scénarios, et l'utilisateur peut voir les

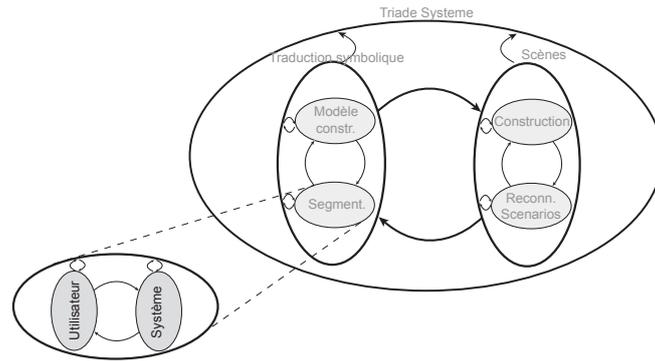


FIG. 3.6.

Intégration de l'utilisateur dans la hiérarchie de triades au niveau le plus inférieur. Les interactions sont situées dans la tâche à réaliser.

conséquences de son action sur tous les niveaux en même temps. En effet, le système prend en compte l'interaction comme une perturbation de la cohérence de son interprétation. Les modifications engendrées pour recouvrer cette cohérence auront des conséquences sur tous les niveaux de l'interprétation. Pour le clinicien, il pourra observer les modifications engendrées par une annotation, la modifier ou la raffiner en fonction du résultat.

Comme il a été présenté dans le chapitre 2, les interactions se font au travers des annotations. Les interactions consistent pour la plupart à ajouter, supprimer ou modifier ses propres annotations ou celles du collaborateur. Certaines interactions utilisent des modes d'interaction "à propos" des annotations. Les interactions au travers des annotations sont un moyen de permettre d'échanger des connaissances tout en conservant les propres représentations de chacun.

Pour le système toutes les interactions sont en fait des perturbations de son environnement (*i.e.* l'environnement des agents contenant les annotations) qui vont le forcer indirectement à modifier son interprétation pour qu'elle soit cohérente avec les nouvelles perturbations introduites.

A.3.2 Interactions par les annotations

L'environnement dans lequel évolue les agents est en fait le même que celui qui mis à disposition de l'utilisateur. Lors de la présentation de l'environnement (*cf.* section A.1.4), nous avons indiqué que nous ne mettons pas les modèles dans cet environnement. En effet, dans le cadre de notre système l'utilisateur n'a pas accès aux modèles construits par la machine² pendant la construction de leur interprétation. L'utilisateur ne fait qu'échanger les annotations qui sont dans l'environnement partagé par les agents, et maintenant avec l'utilisateur.

Les annotations des séries temporelles, *i.e.* l'ajout sur des séries de segments, symboles et scènes permettent à l'utilisateur et à la machine d'échanger des informations de manière symétrique. Les annotations ne sont ici que des marques ancrées dans les séries temporelles mais sans contenu textuel. Les symboles et les scènes annotées se réfèrent aux événements et aux scénarios qui sont les modèles en cours de construction.

L'utilisateur du système réalise une annotation des séries temporelles au moyen de l'interface graphique. Pour cela, l'utilisateur sélectionne un des outils et il l'utilise directement sur les interfaces des séries temporelles pour modifier l'annotation. Les outils qui sont à sa disposition permettent :

- d'ajouter des annotations,
- de modifier des annotations,
- de supprimer des annotations.

Une fois que l'outil adéquat a été sélectionné, l'ajout d'annotations permet à l'utilisateur d'ancrer des

²En pratique, on peut observer les modèles qui sont construits, mais l'utilisateur ne les utilise pas pendant son interprétation.

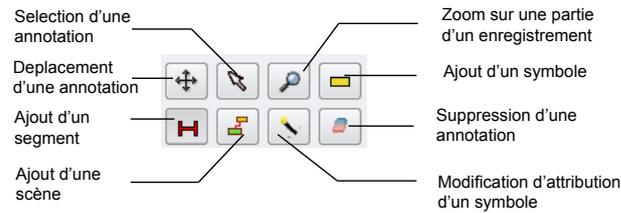


FIG. 3.7. Outils qui peuvent être utilisés pour réaliser l'annotation.

segments ou des symboles sur une série temporelle et des scènes sur un enregistrement.

- Pour l'ajout d'un segment, l'utilisateur pointe le segment d'une série temporelle qu'il souhaite annoter à l'aide de la souris. Une barre horizontale rouge apparaît alors pour notifier l'ajout effectif de l'annotation.
- Pour l'ajout de symboles, l'utilisateur peut préciser au préalable le type d'évènements qu'il souhaite annoter ou bien demander au système de lui proposer lui-même parmi les évènements ceux qui correspondent le mieux une fois qu'il aura sélectionné la période de la série temporelle à associer au symbole.
- Pour l'ajout de scénarios, l'utilisateur doit interagir avec les symboles déjà existants sur un enregistrement. Pour ajouter une scène, l'utilisateur sélectionne l'ensemble des symboles qui doivent constituer la scène. L'utilisateur peut alors se voir proposer par le système des attributions de scénarios, soit le choisir lui-même.

Pour les modifications, l'utilisateur sélectionne une annotation et la modifie. Pour toutes les annotations on peut modifier les informations temporelles (dates et durées) en déplaçant les annotations sur les séries temporelles. Pour les symboles et les scènes, on peut également modifier leur attribution à un évènement ou à un scénario. Pour le système, une modification d'une annotation propre à l'utilisateur se passe comme si l'annotation ancienne avait été supprimée (négation de l'ancienne annotation) et qu'une nouvelle est introduite.

Finalement, l'utilisateur peut supprimer ses propres annotations ou celles du système en utilisant la gomme. Par un simple clique, il peut supprimer les annotations ancrées sur une série.

Les annotations qui sont visibles dans l'interface sont produits à la fois par l'utilisateur et l'interface. Les annotations sont associés à l'identifiant du partenaire qui les a créé. Pour la modification des annotations produites par le système, la nouvelle annotation est associée à l'utilisateur comme producteur. La différence pour le système est dans la suppression des annotations. Les annotations réalisées par le clinicien ne pourront pas être modifiées ou supprimées par le système. Sinon les annotations des deux partenaires sont considérées de la même façon dans la construction de l'interprétation. Sur l'interface, l'utilisateur peut distinguer les annotations du système de celles de l'utilisateur soit en utilisant des couleurs légèrement différentes (moins saturées pour les annotations de la machine) ou bien en utilisant des affichages simultanés séparés des annotations de la machine et de l'utilisateur.

A.3.3 Modification des modèles par les exemples

Interaction pour affiner les modèles d'évènements Une interaction spécifique a été mise en place pour pouvoir affiner la construction des modèles d'évènements. Dans le cas d'une attribution d'un symbole à un évènement qui ne semble pas correspondre à l'utilisateur, on peut utiliser deux modes de modifications :

- le mode *fort* qui force l'introduction d'un exemple dans une classe,
- le mode *faible* qui indique une similarité entre deux exemples ;

Dans le mode fort, l'utilisateur estime sa connaissance sur l'exemple de symbole considéré suffisamment sûre pour forcer l'association à un évènement cible. Si par la suite le système ne juge plus intéressant de conserver l'attribution de ce symbole à un évènement, il pourra très bien modifier cette attribution. Ceci peut conduire l'utilisateur à devoir réaliser plusieurs fois une modification pour le même symbole.

Du point de vue de l'IHM, c'est une limite prévisible du système.

Mais dans certains cas, l'utilisateur n'est pas nécessairement sûr de l'introduction qu'il peut proposer ou même ne pas être capable de nommer l'évènement auquel est associé un segment de séries temporelles. Le mode faible propose donc une interaction inspirée de *Talking Heads* de Steels [201]. Dans les scènes précédentes, l'utilisateur sera certainement en mesure de répondre à une question du type : "est-ce que ce segment est le même que celui-ci?".

L'interaction est réalisée comme suit. Le système propose trois exemples de segments correspondants à trois évènements différents (le contexte du segment peut être consulté). L'utilisateur indique le segment du signal qui lui semble le plus en accord avec le segment dont il cherche à modifier l'évènement correspondant. Alors le système crée un lien virtuel entre les exemples qui va forcer le système à considérer ces deux exemples comme identiques. Si l'utilisateur ne trouve aucune correspondance valable, il peut proposer de lui-même un autre exemple qui lui semble identique.

La solution d'ajout de liens est intermédiaire pour assurer que les interactions réalisées par l'utilisateur seront prises en compte (*cf.* la description de l'algorithme de classification en section C.2) et qu'il n'aura pas à les répéter mais que de son côté la machine reste suffisamment *libre* pour construire ses propres modèles de segments. Les liens qui sont réalisés entre deux exemples sont conservés par la machine. Donc l'utilisateur aura la garantie qu'une fois liés, les deux segments seront considérés comme identiques dans le traitement. On peut noter également que la construction de plusieurs liens permet de grouper plusieurs exemples entre eux. Pour la machine, elle pourra toujours s'approprier ces exemples pour se construire ses propres évènements. Elle aura la seule contrainte de considérer dans leur ensemble des exemples qui sont liés.

Ces deux modes d'interactions permettent au système de raffiner la construction des modèles d'évènements. L'utilisateur peut utiliser au coup par coup l'un ou l'autre des modes. Ils sont possibles grâce à l'implémentation d'un algorithme de classification spécifique dans les *agents de construction de modèles*.

Interaction avec les scénarios et les scènes Des interactions similaires à celles utilisées pour les modèles d'évènements pourraient être envisagées. Dans l'état actuel du système, elles n'ont pas été implémentées pour des raisons pratiques : la mise en place d'algorithmes pour que la machine prenne en compte des liens virtuels entre les scènes nécessite un travail important qui n'a été possible de mener dans cette thèse.

En résumé, les points importants à retenir sur le système sont :

- Le système est un système multi-agents dont l'organisation en hiérarchie de triades permet de définir une dynamique qui va vers la construction de la cohérence entre les productions de différents agents :
 - Cohérence interne du système, *i.e.* entre les productions des agents,
 - Cohérence entre le système et l'utilisateur par les échanges d'annotations;
- Les agents produisent des annotations et des modèles pour :
 - agir dans leur environnement en le modifiant,
 - interagir de manière asynchrone avec les autres agents artificiels qui peuvent être intéressés par ces informations,
 - interagir avec l'utilisateur, puisque les annotations et les données constituent un environnement partagés avec le clinicien.

B Modèles

Nous mettons en évidence dans un premier paragraphe l'importance que revêt le choix d'un modèle pour la réussite pratique de notre système à obtenir une interprétation. Nous expliquons ensuite comment nous avons dirigé ces choix par la limitation de l'introduction d'*a priori*.

Nous introduisons ensuite trois sortes de modèles qui sont utilisés dans notre système. Les **modèles de segments** sont des représentations des segments qui doivent permettre de clarifier la notion de distance entre segments. Les **modèles d'évènements** ont la charge de représenter les caractéristiques d'une classe de segments. Dans notre processus, une classe de segments est associée à un évènement, on les nomme

donc des modèles d'évènements. Enfin, nous clarifions la notion de scénario en proposant un modèle formel. Cette définition nous servira pour l'introduction de la notion d'**explication** que nous utilisons pour l'apprentissage.

B.1 Choix des modèles et des distances

B.1.1 Relation des modèles et distances à la perception

Dans l'introduction au paradigme de l'Enaction, nous avons mis en évidence que la perception était un élément important dans la cognition.

La perception était initialement contrainte par les caractéristiques physiques qui nous mettent en relation avec le monde. Dans le cas de notre interprétation, le monde est composé des données, des annotations et des connaissances qui y sont construites. On peut voir les modèles informatiques utilisés *a priori* comme les organes de perception de la machine des données. C'est à travers ces modèles que la machine sera capable de percevoir et agir dans le monde.

Par conséquent, l'approche Enactive met en évidence d'une part que les résultats de l'interprétation sont entièrement conditionnés aux choix des modèles, et d'autre que part la construction d'un système "Enactif" doit pouvoir adapter, partiellement, ses perceptions, *i.e.* les modèles, au cours de son expérience. Ce sont donc deux contraintes que les modèles choisis doivent respecter.

B.1.2 Introduction d'*a priori* par les modèles

Le choix des modèles est une chose importante dans la réalisation d'un système qui travaille avec des abstractions de données. De ce choix dépend l'obtention de résultats pratiques. La modélisation des données a pour effet de mettre en évidence les caractéristiques des données, mais également de faire disparaître toutes les autres informations.

Le choix des modèles est généralement lié à l'application. Lorsqu'un expert est capable de donner les caractéristiques qu'il faut observer, alors on peut en dériver les modèles qui permettront d'obtenir les mêmes résultats que l'expert³. Mais dans le cas où un expert ne peut pas donner précisément ces caractéristiques, par manque de connaissance ou par manque de formalisation, il faut alors proposer des modèles suffisamment *larges* pour espérer couvrir les caractéristiques qui permettront de focaliser sur une information intéressante en limitant la perte.

Ceci peut se ramener à un choix d'introduction d'*a priori* dans les processus d'abstraction avec le compromis suivant : Si on introduit beaucoup d'*a priori*, on contraint d'autant plus le processus assurant une abstraction en accord avec les modèles mais qui a peu de chance de découvrir de nouveaux éléments intéressants, tandis que si on en introduit peu, on facilite la découverte de nouveaux éléments, mais en risquant d'avoir une abstraction peu sensée.

B.1.3 L'importance des distances

Nous nous intéressons plus précisément à la construction d'un modèle *informatique*. De tels modèles sont par essence des objets calculables. En premier lieu, dans nos abstractions c'est la notion de distance qui est la plus utilisée :

- Distance entre les modèles et les objets
 - Distance entre le modèle et les données pour trouver l'évènement correspondant à un modèle d'évènement pour les *agents de symbolisation*,
 - Distance entre les scénarios et les séries symboliques pour identifier les scènes,
- Distance entre les objets
 - Distance entre les segments modélisés pour construire des classes pour l'*agent de construction de modèles*,
 - Distance entre les scènes pour les *agents d'apprentissage de scénarios*,

Nous souhaitons ici mettre en évidence l'importance que peut prendre une distance dans nos traitements des données. Ils sont en effet une introduction (souvent implicite) d'une connaissance *a priori*.

³On rappelle ici l'hypothèse réalisée dans le chapitre 1 qui a été de supposer que l'information nécessaire était dans les données. Ici, ce peut être un nouveau facteur limitant pour la production d'un modèle adéquate aux connaissances *a priori*.

Par exemple, si on prend un jeu de données que l'on souhaite classer, l'objectif de la classification est de maximiser les distances entre les exemples inter-classe et minimiser les distances intra-classes. Alors le résultat de la classification est déterminé par la distance choisie. Les algorithmes ne sont que des heuristiques pour atteindre cet optimum.

C'est pourquoi il nous semble très important d'être conscient de l'importance d'un tel choix, et que si on souhaite limiter l'introduction d'*a priori* dans nos abstractions, il faut essayer de relâcher les contraintes imposées par le choix des distances.

B.2 Modèle de segments

Le modèle de segments doit permettre de spécifier les informations significatives dans un segment d'une série temporelle. L'information significative est déterminée par les critères sur lesquels le système effectue la comparaison des segments. Il existe trois distances usuelles qui ne comparent pas les segments sur les mêmes critères.

B.2.1 Distances usuelles entre segments

On prend ici comme exemple la comparaison des segments de la figure 3.8. Trois transformations ont été réalisées à partir du segment original :

1. les décalages temporels à l'origine ;
2. les dilatations temporelles ;
3. une translation selon les ordonnées ;

Nous présentons dans cette section trois distances usuelles : la distance euclidienne, la Dynamic Time Warping (DTW) [108] et la longest common subsequence (LCSS) [70]. Afin de mettre en évidence que des distances différentes ne permettent pas d'avoir une même vue sur ces segments, on confronte qualitativement les distances obtenues pour les exemples de la figure 3.8.

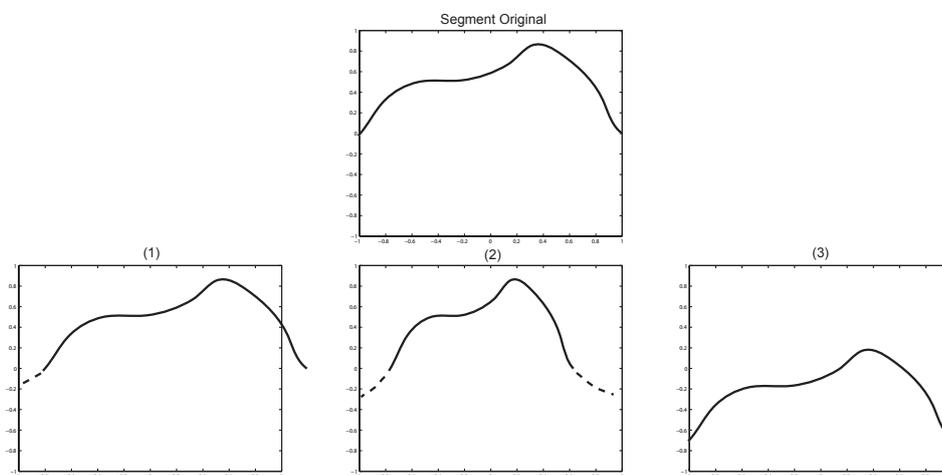


FIG. 3.8.

Les trois segments (1), (2) et (3) sont-ils identiques au segment original? Les trois segments ont été obtenus (1) par décalage temporel, (2) par dilatation temporelle, (3) par translation selon les ordonnées. La comparaison de deux segments est caractéristique du modèle qui fixe les informations importantes d'un segment.

Distance Euclidienne La distance Euclidienne ne peut prendre en compte que des segments de même taille. Comme le montre la figure 3.9, elle somme les différences de valeurs point à point.

$$Euclidian(S^1, S^2) = \sum_{x=0}^l (s_x^1 - s_x^2)^2$$

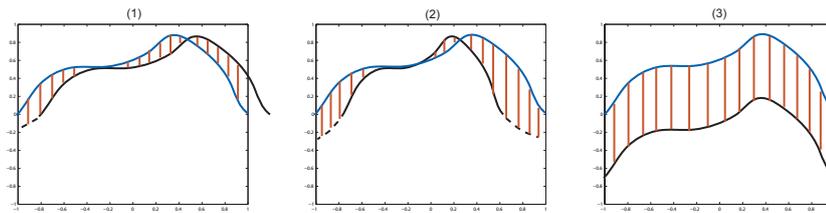


FIG. 3.9.
Exemples de mesure de distance Euclidienne entre deux segments.

Pour toutes les transformations du segment original qui ont été données, la distance euclidienne donne une grande valeur et les exemples peuvent alors ne pas être considérés comme identiques⁴. La distance euclidienne est utilisée pour des données dans lesquelles tous les segments sont de même taille et pour lesquelles les segments obtenus par les transformations précédentes modifient le “sens” du segment.

Dynamique Time Warping Soient deux segments $X = \{x_1, x_2, \dots, x_n\}$ et $Y = \{y_1, y_2, \dots, y_m\}$ non nécessairement de même taille. L’algorithme de DTW propose de construire deux sous-séquences croissantes de longueur k , i_1, \dots, i_k et j_1, \dots, j_k qui minimisent la mesure suivante :

$$\sum_{l=1}^k (X_{i_l} - Y_{j_l})^2$$

Cette quantité donne la distance entre les deux segments. La solution du problème peut être représentée graphiquement sous la forme d’un chemin dans la matrice des distances points à points (Matrice de coût). L’idée de la DTW est de formuler ce problème sous la forme d’un algorithme de programmation dynamique qui permet de calculer la mesure dans un temps raisonnable (non proportionnel à la taille de la matrice $n \times m$).

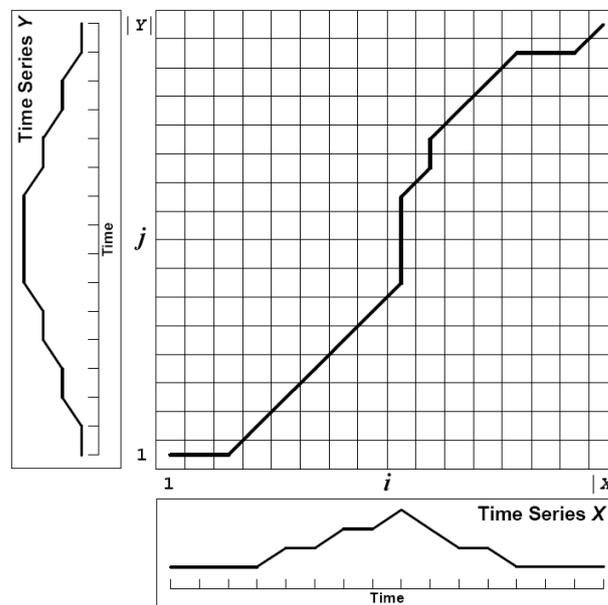


FIG. 3.10.
Matrice de coût et chemin optimal obtenu entre deux séries temporelles (Figure extraite de [181])

La figure 3.11 illustre le type de comparaisons qui peut être réalisé par une DTW. La mesure entre les deux est la somme des longueurs des segments rouges. On peut constater que la DTW a la propriété de donner des mesures faibles pour des segments similaires avec des dilatations et des translations temporelles

⁴Pour savoir si deux segments sont identiques, il faut se fixer un seuil *a priori* de similarité.

(contrairement à la distance Euclidienne). La mesure entre des segments étant uniquement translatés reste du même ordre que pour la distance Euclidienne.

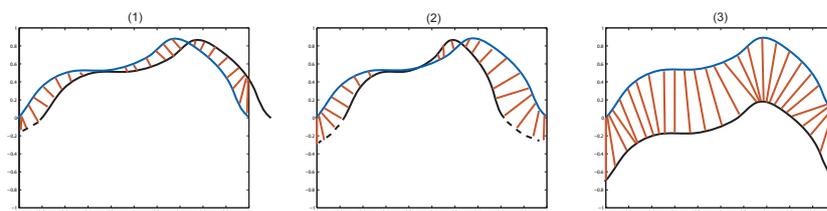


FIG. 3.11.
Exemples de mesure de distance DTW entre deux segments.

La DTW a beaucoup été utilisée en reconnaissance de la parole pour estimer une distance en prenant en compte des échelles différentes. Des variantes permettent d'introduire des améliorations dans la recherche d'un chemin optimal en ajoutant des contraintes à la matrice de coût [123]. Gaudin et *al.* proposent quant à eux une version adaptable de ces contraintes [101]. Enfin, une récente amélioration de l'algorithme, FastDTW, a été proposée par Salvador et Chan [181] en utilisant une approche multi-niveaux.

Longest Common Sub-Sequence Das et *al.* [70] définissent une mesure de similarité entre les deux segments $X = \{x_1, x_2, \dots, x_n\}$ et $Y = \{y_1, y_2, \dots, y_m\}$ non nécessairement de même taille. X et Y sont similaires s'il existe deux sous-séquences d'indices croissants $i_1, \dots, i_{\gamma n}$ et $j_1, \dots, j_{\gamma n}$, telles que $X' = \{x_{i_1}, x_{i_2}, \dots, x_{i_{\gamma n}}\}$ et $Y' = \{y_{j_1}, y_{j_2}, \dots, y_{j_{\gamma n}}\}$ répondent à la contrainte suivante :

$$\forall k, 1 \leq k \leq \gamma n, \frac{y_{j_k}}{1 + \epsilon} \leq a \cdot x_{i_k} + b \leq \frac{y_{j_k}}{1 - \epsilon}$$

γ et ϵ sont deux paramètres entre 0 et 1. γ fixe la longueur de la sous-séquence utilisée, et ϵ définit la signification de sous-séquence commune. Dès lors, l'algorithme LCSS est un algorithme de programmation dynamique qui permet de trouver la plus grande sous-séquence commune aux deux segments. La distance est ensuite obtenue à partir de ces sous-séquences en sommant les distances point à point.

Le résultat sur notre exemple de l'association entre les points des segments aurait été assez semblable aux résultats de la DTW. Pour illustrer les différences entre les deux algorithmes, nous reprenons sur la figure 3.12 l'exemple de Keogh et *al.* [129] pour laquelle la différence est évidente.

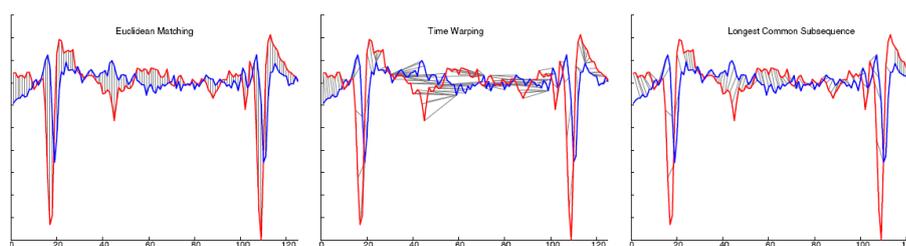


FIG. 3.12.
Comparaison entre les distances euclidienne, DTW et LCSS sur un exemple avantageux pour la LCSS (Figure extraite de [129]).

Cependant, sur notre exemple si les associations avaient été les mêmes, les mesures auraient été différentes pour le troisième exemple. Dans la mesure où l'algorithme LCSS prend en compte une transformation linéaire possible entre les valeurs des segments, la translation selon les ordonnées peut être prise en compte de sorte à obtenir pour cette comparaison également une mesure faible. Ainsi la méthode identifie les quatre exemples comme similaires.

En conclusion, les trois différentes distances ne permettent pas de comparer les segments de la même manière de sorte que si on les utilisaient dans le même algorithme de classification, ils donneraient des résultats différents. De plus, la DTW et la LCSS mettent en évidence la difficulté de comparer des

segments lorsqu'on souhaite prendre en compte des dilatations des segments [62, 63]. Cette difficulté de comparer des segments de manière robuste est accrue si il y a du bruit dans les données.

B.2.2 Choix *a priori* d'une distance

Le choix d'un modèle de segment soulève de problème du choix d'un critère de similarité en deux segments. Si on dispose de ressemblances et de dissemblances *a priori* des segments, alors il faut que les caractéristiques puissent les mettre en évidence. Prenons de nouveau l'exemple de la figure 3.8. Dans quelle mesure deux segments peuvent être considérés comme identiques? Doit-on prendre en compte ou ignorer les différences de dilatation ou de décalage? Sans connaissance *a priori* sur les critères de comparaison des segments, on ne peut pas répondre à cette question. Pour certaines applications ces critères sont sans importance et pour d'autres ils en ont beaucoup.

Nous comptons acquérir cette connaissance au cours du processus d'interprétation des séries temporelles. En effet, si initialement, la machine part avec sa propre idée de segments similaires, les interactions avec le clinicien et sa propre expérience du traitement des données doivent lui permettre de raffiner ses comparaisons au cours du processus pour qu'elles correspondent mieux à son expérience (*i.e.* le fait qu'on lui ait indiqué que les exemples doivent être similaires).

De plus, ne pas imposer une distance permet de limiter l'introduction d'une connaissance *a priori*. Le système conserve ainsi une certaine autonomie de la construction des événements.

Nous devons donc proposer un modèle de segment qui va permettre d'utiliser une distance adaptable au cours de l'interprétation. Le processus d'adaptation de la distance sera présenté dans la section C.2.

B.2.3 Proposition d'un modèle de segment

Les distances qui ont été présentées précédemment (euclidienne, DWT et LCSS) s'appuient sur une caractérisation de la similarité fondée sur la ressemblance de forme (visuelle pour un expert humain). Mais ce critère de comparaison n'est pas nécessairement le plus adéquat : peut être que l'information significative pour comparer deux segments se trouve dans la seule valeur extrême du segment, sa tendance globale (en particulier, on a déjà pu voir que les méthodes d'abstraction temporelles de séries temporelles médicales s'appuient beaucoup sur les tendances), ou une autre caractéristique d'un segment ...

Verduijn [219] propose d'utiliser des représentations vectorielles des segments. Un segment est alors représenté par un vecteur de grande dimension contenant un grand nombre de descripteurs du segment. En ayant à disposition une grande quantité de descripteurs du segment, le processus d'adaptation de la distance va permettre de focaliser la comparaison uniquement sur ceux qui sont importants pour les données traitées. De plus, en utilisant une représentation vectorielle de segments, on s'affranchit des contraintes liées à la dimension temporelle puisque celle-ci n'intervient que comme un paramètre supplémentaire.

Nous proposons une représentation d'un segment selon 10 critères quantitatifs et les plus objectifs possibles. Cependant, en fonction des applications, il peut être ajouté de nouvelles dimensions pour affiner une description des segments. Verduijn et *al.* [219] disposent de près de 300 caractéristiques des segments.

	Durée	
	Moyenne	
	Variance	
	Maximum	
	Minimum	
	Tendance globale	
	Tendance de la première moitié du segment	
	Tendance de la second moitié du segment	
	Dérivée maximum	
	Quantité de bruit	

De sorte à avoir une égale importance entre les différents critères, on modifie légèrement ce modèle en indiquant que chacune des valeurs est normalisée entre 0 et 1. Les vecteurs de normalisation sont donnés *a priori* sur la connaissance espérée des valeurs maximales et minimales de chacun des paramètres.

Adaptation de la distance Le choix du modèle nous aide dans la solution pour une distance adaptative. En effet, les segments étant des vecteurs, une distance entre les segments sera une distance euclidienne pondérée par des valeurs entre 0 et 1. Et la pondération sera le reflet de l'importance des critères sur lesquels comparer les segments. Lorsque un critère sera pondéré avec une valeur proche de 0, quelque soit la différence la mesure obtenue sera faible. Le critère n'aura que peu d'importance. En revanche, si la pondération est proche de 1, la moindre différence entre les deux valeurs aura des conséquences importantes sur la similarité globale.

Soient s_1 et s_2 deux segments, S_1 et S_2 sont les vecteurs qui représentent ces segments. Et soit D la matrice diagonale des poids adaptables de la distance. On a alors la distance entre deux segments qui est donnée par la formule :

$$d(s_1, s_2) = \sqrt{S_1^t \cdot D \cdot S_2}$$

La normalisation réalisée du modèle permet d'avoir une égalité d'importance entre les critères. Si par exemple un critère évoluait entre des valeurs de 10^2 et 10^3 , alors qu'un autre critère évoluait entre des valeurs de 10^{-3} et 10^{-2} , même si la pondération du premier était faible par rapport au second (*e.g.* 2 fois plus petit) il resterait artificiellement un critère important dans la similarité.

Construction d'une "perception" des segments Du point de vue de l'Enaction, on cherche à mettre ici en place le support à un mécanisme qui va permettre au système de se construire sa "perception" du système en extrayant les caractéristiques importantes sur lesquels il faut comparer deux segments. Le vecteur de critères représente les capacités de perception du système et les poids adaptables de la distance mettent en avant ceux qui sont les plus importants et qui auront été extraits de l'expérience passée du système grâce à l'algorithme de classification présenté dans la section C.2.

B.3 Modèle d'évènement

Un modèle d'évènement est associé dans notre système à un ensemble de parties de séries temporelles (segments), qui sont définies par les symboles qui annotent l'évènement. Un évènement peut donc être associé à un ensemble de segments qui sont des instances de cet évènement. De manière non-représentationnelle, on a donc ainsi une trace de l'expérience réalisée par le système à propos d'un évènement.

Le modèle d'évènement est construit à partir de cet ensemble de segments pour être utilisé dans des *agents de segmentation*. Plus précisément, le modèle d'évènement est construit à partir des représentations des segments sous forme vectorielle.

Deux approches sont alors possibles pour représenter un ensemble de valeurs, on peut construire un prototype à partir des données qui servira de référence à l'ensemble d'une classe de segments ou bien tenir compte de la variabilité des exemples de la classe.

Nous proposons d'utiliser un modèle d'évènement qui dérive du modèle de segment et qui prenne en compte la variabilité des segments d'une classe. Pour cela, le modèle d'évènement est constitué de deux vecteurs de même dimension que les vecteurs du modèle de segments et une matrice de distance D^E . Le premier V_{max}^E contient les valeurs maximales rencontrées dans les exemples et le second V_{min}^E les valeurs minimales.

Pour l'*agent de segmentation* on a besoin de définir une distance entre un segment et le modèle. C'est à partir de cette distance que l'agent pourra décider si un segment correspond bien à son modèle ou non. Pour évaluer la distance d'un segment au modèle d'évènement, on modélise tout d'abord le segment S sous sa forme vectorielle V_S . La distance est alors calculée ainsi :

$$dist(V, Modele) = \sqrt{C^t D^E C}$$

avec C qui est un vecteur dont les composantes sont définies ainsi :

$$C_i = \begin{cases} 0 & \text{si } V_{min,i}^E \leq V_i \leq V_{max,i}^E \\ \max(V_{max,i}^E - V_i, V_i - V_{min,i}^E) & \text{sinon} \end{cases}$$

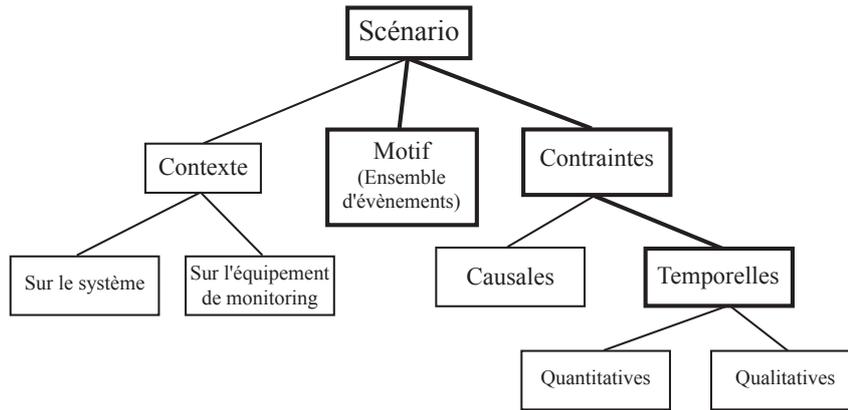


FIG. 3.13.

Formalisme de scénario : un scénario est principalement constitué d'un ensemble d'évènements et de contraintes sur ces évènements.

Le vecteur C nous indique que si une des valeurs se trouve dans la fourchette des valeurs d'un critère acceptable pour le modèle, alors la distance n'est pas pénalisée. En revanche, s'il s'en écarte, la distance augmente avec sa distance aux bornes de la fourchette autorisée. La matrice diagonale de la distance déjà utilisée pour la comparaison de segment permet de pondérer cette distance uniquement sur les critères significatifs.

Il est à noter que chaque modèle d'évènement dispose d'une distance adaptée propre D^E . Les sciences cognitives nous enseignent que deux exemples ne sont pas comparés de la même façon s'ils appartiennent à la même classe ou non (Goldstone et al. [103]). On part donc du principe que pour savoir si un objet correspond à un type d'objet on ne le compare pas systématiquement sur les mêmes critères. Par exemple, si je connais les types d'objets *chaise*, *table* et qu'on me présente un tabouret. Si je cherche à savoir si un tabouret est plutôt une *chaise* ou une *table*, je ne considérerai pas les mêmes critères. On fait ici de même pour les évènements : pour savoir si un segment correspond à un évènement, on le compare sur les critères signifiants de l'évènements.

B.4 Scénarios et explications

Précisons maintenant les modèles de ce que nous appelons des scénarios. On trouve parfois le terme de chronique [83] qui recouvre la même idée. Ensuite, nous présentons la notion d'explication que nous utiliseront dans notre méthode pour permettre de distribuer à plusieurs *agents d'apprentissage* la construction de l'ensemble des scénarios.

B.4.1 Modèle de scénarios

La figure 3.13 illustre un formalisme général dans lequel on peut définir un scénario.

La partie principale d'un scénario est l'ensemble d'évènements (motif) contraint par des relations temporelles. Les évènements sont associés à un ensemble de contraintes :

- contraintes temporelles quantitatives (*e.g.* entre 1s. et 2s., maximum 30s., ...) sur leurs dates, leurs durées ou les intervalles de temps entre les évènements et/ou qualitatives (*e.g.* avant, pendant, ...),
- contraintes causales (relations de conséquences entre évènements).

L'ensemble de ces contraintes doivent être cohérentes (ne pas se contredire entre elles) mais pas nécessairement complète (*i.e.* fixer toutes les dates et durées des évènements).

Le contexte de scénario permet d'informer les conditions générales du système qui correspondent ce type de scénarios. Si on prend l'exemple de patients monitorés, les scénarios de patients ayant subit un traumatisme est à distinguer du cas de patients sains. C'est donc une information contextuelle. Le contexte peut aussi contenir des informations sur la récupération des données (disponibilité des capteurs, ...).

Dans notre système, un modèle de scénario est simplement un ensemble d'évènements liés par des

contraintes temporelles quantitatives définies par des fourchettes de dates et de durées et événements. Un scénario peut ainsi se représenter comme un ensemble de triplet (*Evenement, Date, Duree*). Il peut alors être représenté sous la forme d'un graphe temporel [82], *i.e.* un graphe orienté dont les places qui représentent les événements sont qualifiées par une durée typique de l'évènement et dont les transitions représentent une durée typique entre deux événements (définies par les dates des événements).

B.4.2 Explication

Une "explication" est un type de scénario particulier. C'est en fait une forme de règle logique dans laquelle la prémisse est composée d'évènements liés par des relations temporelles à la conclusion comportant un unique évènement (l'évènement à expliquer). Dans une explication nous ne considérons que des relations temporelles quantitatives sur les dates et les durées des événements et pas de relations qualitatives (*e.g.* avant, après, ...). Toutes les dates sont exprimées par rapport à la date de début de l'évènement à expliquer. De plus, les contraintes temporelles de notre modèle de scénarios sont des dates typiques⁵.

Sur la figure 3.14, il est montré comment passer d'un scénario sous la forme de graphe à une explication. Dans le paragraphe C.3, d'autres modèles de scénarios sont présentés. De même que pour les modèles de segments, les représentations ne sont pas équivalentes. L'utilisation de la représentation sous la forme d'explication nous permet de construire des exemples d'une explication à partir des séries symboliques et ainsi de faire de l'apprentissage supervisé des scénarios.

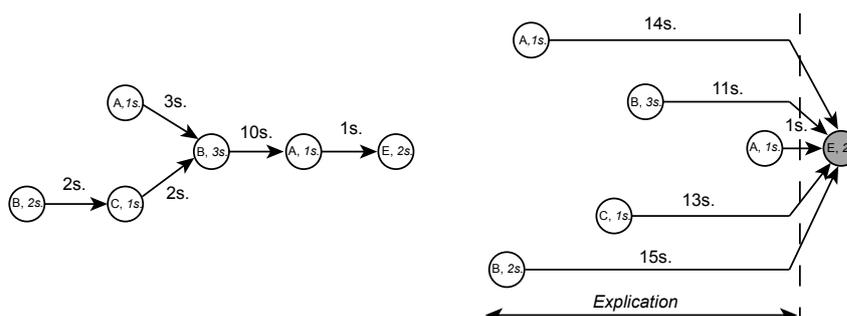


FIG. 3.14.

Représentation d'un scénario sous la forme d'une explication. À gauche, la représentation d'un scénario sous la forme d'un graphe temporel et à droite la représentation du même scénario sous la forme d'une explication. L'évènement le plus tardif E dans le scénario est devenu l'évènement à expliquer par l'explication.

Pour reconnaître une scène dans un ensemble de symboles, il faut d'une part identifier le motif dans l'ensemble des symboles et ensuite l'adéquation des symboles aux durées typiques du scénario permet de donner une mesure de similarité. Si cette mesure est inférieure à un seuil, alors on peut associer l'ensemble de symboles correspondant au motif à une scène correspondante à ce scénario.

L'adéquation est calculée en projetant le scénario sur l'ensemble des symboles. La mesure de similarité est alors calculée comme le support commun entre le scénario (incluant le symbole de l'évènement à expliquer) et l'ensemble des symboles. Le support commun est calculé comme la somme des parties communes entre symboles identiques. Cette similarité est utilisée en particulier pour les *agents de reconnaissance de scénarios* qui annotent des scènes correspondantes à un scénario.

Soient un scénario $Scen = \{(S_i^{Scen}, date_i^{Scen}, duree_i^{Scen})\}_{i \in \mathbb{N}_n}$ et un ensemble de symboles $S = \{(S_i, date_i, duree_i)\}_{i \in \mathbb{N}_m}$, la mesure de similarité entre le scénario et les symboles, illustrée par la figure 3.15, est calculée ainsi :

$$d(Scen, S) = \sum_{k=1}^n \sum_{l=1}^m \|(S_k^{Scen}, date_k^{Scen}, duree_k^{Scen}), (S_l, date_l, duree_l)\|$$

⁵On aurait pu également utiliser des intervalles de valeurs acceptables.

avec la mesure de similarité entre les symboles définie par :

$$\|(S^1, date^1, duree^1), (S^2, date^2, duree^2)\| = \begin{cases} 0 & \text{si } S^1 \neq S^2 \\ \frac{\min(date^1+duree^1, date^2+duree^2) - \max(date^1, date^2)}{\max(duree^1, duree^2)} & \text{sinon} \end{cases}$$

Le recouvrement de deux symboles S^1 et S^2 (formule donnée dans la similarité de deux symboles) est donc une mesure entre 0 et 1 en pondérant par la longueur maximum des deux segments.

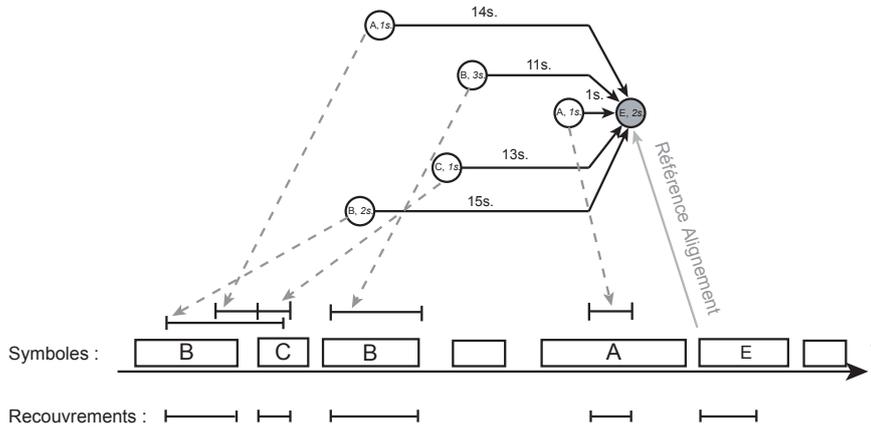


FIG. 3.15.

Calcul d'une similarité entre un ensemble de symboles et un scénario. En haut, le scénario est projeté sur l'ensemble de symboles en alignant les dates sur l'évènement E qui est l'évènement à expliquer. Les segments au dessus des symboles correspondent aux dates et durées reconstituées pour chacun des symboles. Les recouvrements entre ces segments et les symboles de même type sont représentés par les segments tout en bas de la figure. La mesure de similarité est donnée par la somme des durées de ces segments.

Le terme d'*explication* est à prendre avec un sens faible. En effet, on peut se demander si une explication est un scénario qui permet d'*expliquer* les occurrences d'un évènement donné. Formellement, une explication n'a pas de dimension causale. Ce qui va effectivement être observé ce sont des co-occurrences entre l'apparition d'un évènement et des éléments qui le précèdent. Par exemple, s'il n'existe que deux types d'évènement A et B qui se répètent alternativement ($ABABABA$), il est difficile de dire en toute objectivité que A explique B et que B explique A . La dimension explicative ne peut venir que de l'interpréteur clinicien dans le cadre d'une compréhension physiologique des phénomènes qui sont mis en évidence par ces scénarios.

C Principaux algorithmes de traitements des données

C.1 Segmentation des séries temporelles

On détaille dans cette section le traitement d'un unique *agent de segmentation*. Son but est d'annoter les séries temporelles à l'aide de segments. Pour cela, l'agent cherche les segments qui correspondent au mieux avec un modèle de segment qui lui est spécifique et qu'il embarque. L'*agent de segmentation* applique donc un algorithme de *pattern matching* aux séries temporelles.

La segmentation des séries temporelles est le résultat du traitement de plusieurs *agents de segmentation*. Parmi les algorithmes de segmentation dont nous présentons au préalable quelques approches classiques, nous retenons une solution utilisant des fenêtres flottantes pour retrouver dans les séries les segments correspondant aux modèles des évènements.

C.1.1 Méthodes de segmentation de séries temporelles

Le problème général de la segmentation peut se poser ainsi. Soient $S = (s_i)_{i \in \mathbb{N}_n}$ une série temporelle et $E_S = ([s_{b_i}, s_{f_i}])_{i \in \mathbb{N}_p}$ une segmentation de S en p segments.

La segmentation consiste à construire E_S à partir de S de sorte à minimiser une fonction⁶ $F(S, E_S)$ entre S et E_S . En fonction de la méthode cette fonction à minimiser est plus ou moins explicitement utilisée.

Des contraintes supplémentaires à la segmentation peuvent être ajoutées soit par une connaissance *a priori* du résultat attendu, soit pour des raisons algorithmiques permettant de réaliser ou accélérer la segmentation. Lorsqu'on impose que tous les points de la série S appartiennent à au moins un segment, alors on cherche un *recouvrement* de la série par la segmentation. On peut également uniquement chercher des segmentations dans lesquelles les segments sont disjoints. Si on a ces deux contraintes à la fois, la segmentation consiste à construire une *partition* de la série. On a alors $b_1 = 1$, $f_p = n$ et $\forall i, f_i = b_{i+1}$. D'autres contraintes peuvent être imposées comme le nombre de segments p qui constituent la segmentation, ou bien le fait de pouvoir réaliser la segmentation en ligne.

Keogh et *al.* [128] distinguent trois types de méthodes de segmentation de séries temporelles : les méthodes ascendantes, les méthodes descendantes et les méthodes utilisant des fenêtres flottantes. Nous ajoutons à ces méthodes celles qui permettent la détection en ligne de ruptures.

Les méthodes de segmentation ascendantes commencent par proposer une segmentation très fine d'une série temporelle et raffine la segmentation par la fusion de segments voisins en un segment plus gros, homogène. Charbonnier [56] propose une méthode de segmentation en ligne de fusion entre segments approximatés par des fonctions linéaires. Les valeurs qui arrivent progressivement sont agglomérées progressivement au segment en cours de construction et lorsque l'erreur d'approximation passe sous un seuil, un nouveau segment est créé. L'abstraction verticale définie de Shahar et Musen [190] est une méthode pour agglomérer des points grâce à une base de connaissance qui permet de spécifier comment ces agglomérations peuvent être réalisées. En fonction de la nature des séries temporelles, les agglomérations ne sont pas nécessairement les mêmes.

L'approche descendante fait exactement l'inverse, *i.e.* que la segmentation initiale est très grossière mais est progressivement raffinée. Ce type de méthode nécessite d'avoir à disposition l'intégralité de la série temporelle. La méthode de Sharshar et *al.* [191] permet de détecter dans des signaux de SpO_2 des segments qui relèvent d'un comportement normal, transitoire ou perturbé. L'algorithme se base uniquement sur des critères statistiques (indices de Gaussianité) calculées à partir de l'ensemble des données et aide à qualifier les points de la série. Les points similaires sont ensuite regroupés. Himberg et *al.* [113] proposent une méthode de segmentation de signaux audio provenant de téléphones portables afin d'identifier les changements de contextes de l'appel. L'algorithme se base sur une méthode de programmation dynamique. La somme des variances de chaque segment est utilisée comme fonction de coût à minimiser par l'algorithme en positionnant au mieux k frontières qui définissent une *partition* de la série. Le nombre de frontière k est fixé *a priori*.

Les approches de détection de ruptures cherche à construire les frontières des segments. Contrairement, à la méthode précédente qui construit des segments sur la base de l'homogénéité des données entre les segments, les méthodes de détection de ruptures cherche à mettre en évidence les changements dans les données. Ce type de méthode conduit nécessairement à la production d'une partition de la série. Nitanda et *al.* [160] cherchent à segmenter le flux audio d'une radio pour détecter les silences, les moments où il y a de la musique et les moments où des gens parlent. Pour cela, ils s'intéressent aux coupures qu'ils détectent en comparant point à point les caractéristiques du signal dans une fenêtre temporelle précédent l'instant et celle qui suit l'instant. Lorsque ces caractéristiques (moyenne et variance des données) sont très différentes, un changement est détecté. Le même algorithme est utilisé par Desobry et *al.* [78] avec une comparaison des fenêtres sur la base des estimateurs de densités de distribution des données.

Finalement les méthodes à base de fenêtre flottante utilisent une approche de reconnaissance d'un *motif temporel* (*pattern matching* ou *pattern recognition* [168]). Les méthodes à base de fenêtre flottante ne visent pas la construction d'une partition d'une série temporelle mais uniquement l'identification des segments "intéressants" dans les données. Le principe général de ces méthodes est d'utiliser une fenêtre d'une taille fixe qui parcourt la série temporelle : on passe ainsi en revue successivement tous les segments possibles de la taille de la fenêtre. Pour chaque fenêtre, on compare les données au(x) modèle(s) recherché(s) et si la mesure permet d'estimer la ressemblance entre les données et la fenêtre alors un nouveau segment a été identifié. Toute la difficulté de la méthode réside dans le choix correct de la

⁶La segmentation peut également consister à maximiser un critère, comme un critère de vraisemblance, mais on peut alors se ramener à une fonction à minimiser.

distance entre le modèle et les données. Chiu et al. [60] proposent un algorithme qui permet de retrouver des motifs similaires en utilisant une comparaison à partir de la représentation SAX [129].

De manière transversale, on peut distinguer dans ces approches des méthodes qui utilisent des modèles riches qui nécessitent une connaissance formalisée des segments recherchés ou des méthodes qui au contraire utilisent peu de connaissance formalisée (*a priori* ou construite) et qui s'appuient sur les critères de comparaison les plus génériques possibles.

Le tableau 3.2 propose une classification des méthodologies précédemment citées. Les méthodes de détection de rupture n'utilisent que peu de connaissance formalisée, tandis que les méthodes utilisant des fenêtres flottantes utilisent nécessairement beaucoup de connaissance pour définir les modèles et les distances. Pour les autres approches (ascendantes et descendantes) l'utilisation de connaissance varie.

Utilisation de connaissance <i>a priori</i>	Peu	Beaucoup
Ascendante	Charbonnier et al. [56]	Shahar et Musen [190]
Descendante	Himberg et al. [113]	Sharshar et al. [191]
Ruptures	Desobry et al. [78], Nitanda et al. [160]	
Fenêtres flottantes		Chiu et al. [60]

TABLE 3.2.

Classification des méthodes de segmentation par type d'approche et par utilisation de connaissance *a priori*.

C.1.2 Segmentation par reconnaissance de modèles

L'objectif de la segmentation est de révéler des parties des signaux qui potentiellement correspondent à des événements intéressants pour comprendre l'enregistrement. En particulier, la réalisation d'une partition des séries n'est pas intéressante pour l'interprétation de séries temporelles longues. En premier lieu, les segments intéressants pour comprendre l'enregistrement sont ceux qui correspondent à des événements connus comme ceux qui ont déjà été construits dans le cadre d'interprétation antérieures et qui sont représentés par les modèles d'événements des *agents de construction de modèles*.

Donc nous utilisons une méthode nous permettant de retrouver des modèles d'événements qui sont construits par les *agents de construction de modèles* pour mettre en évidence les segments intéressants. L'approche qui nous semble la plus adéquate est donc celle d'un algorithme à base de fenêtre flottante.

La méthode de segmentation est illustrée par la figure 3.16. Il s'agit d'un algorithme de parcours d'une série temporelle avec une fenêtre flottante pour retrouver des segments similaires au modèle. C'est-à-dire qu'on déplace progressivement une fenêtre temporelle (en vert sur la figure) dont la taille est définie par le modèle de segment. À chaque position, on calcule alors la distance entre le segment délimité par la fenêtre et le modèle dont on cherche les occurrences (*cf.* section B.3 pour la distance). Si cette distance est en dessous d'un certain seuil (fixé par l'utilisateur) on a alors trouvé un segment potentiel.

Deux difficultés se présentent alors :

- On cherche des segments qui ne se superposent pas. Or en pratique, on peut constater que si une fenêtre correspond au modèle, la fenêtre suivante (assez peu changée puisqu'en effectuant un décalage d'une position on a uniquement changé deux valeurs) aura une forte probabilité de correspondre également. Avec le contrainte de non-superposition, il faut arriver à choisir la meilleure.
- Nos modèles tiennent compte d'une certaine flexibilité dans la taille des segments acceptables (les tailles sont entre les valeurs minimum et maximum). Il faut donc que l'algorithme utilisé essaye également différentes tailles de fenêtre.

Pour résoudre ces deux problèmes, notre algorithme contient deux phases. Dans un premier temps, on calcule pour toutes les positions et pour toutes les tailles de fenêtre acceptables la distance du segment au modèle. Ces résultats sont accumulés dans une matrice à deux dimensions (position dans la série du début de la fenêtre et sa taille) : la matrice de similarités (*cf.* figure 3.16). Dans une seconde phase, on recherche les segments qui sont acceptables dans cette matrice. Pour cela, on procède itérativement de la sorte :

1. Trouver la plus petite valeur dans la matrice de similarité ;

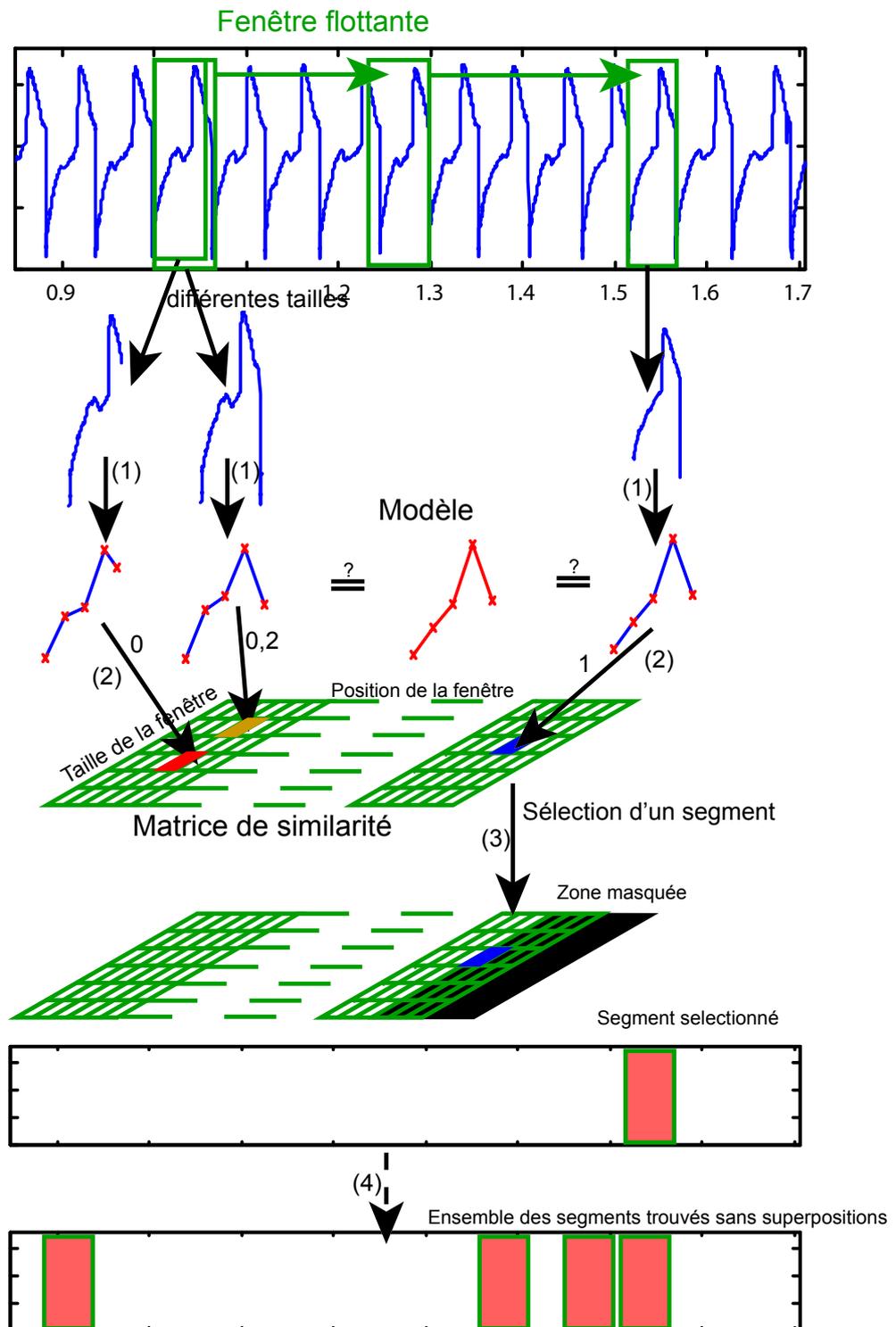


FIG. 3.16. Méthode de segmentation des séries temporelles.

2. Si cette valeur est supérieure au seuil de similarité, on arrête ;
3. Passer à l'infini toutes les valeurs de la matrices qui correspondent à des segments pouvant être superposés au segment choisi (zone masquée) ;
4. Construire le segment correspondant (définit la position et la taille de la fenêtre indiquée par les coordonnées dans la matrice de similarité)
5. Retourner à l'étape 1.

Le masquage de zone *a priori* permet d'accélérer le traitement de la matrice en éliminant à chaque sélection de segment un grand nombre de possibilités pour les explorations futures. Dans la figure 3.17, on illustre en gris les éléments de la matrice qu'il n'est plus nécessaire de parcourir à l'avenir si le segment choisi est à la position (7, 5) dans la matrice de similarité (*i.e.* fenêtre de taille 5 à la position 7 dans la série). Dans la mesure où la matrice est explorée en commençant par la plus petite valeur de similarité, on est assuré qu'il n'y a pas de meilleurs segments potentiels dans ceux qui sont masqués et qui ne seront plus explorés.

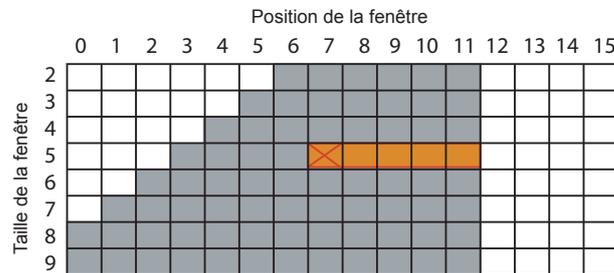


FIG. 3.17. Illustration du masquage d'une zone (en gris) dans la matrice de similarité lorsqu'un segment a été sélectionné (croix).

La contrainte de non-superposition des segments imposée par notre algorithme est valable lorsqu'on recherche des segments de même nature, *i.e.* correspondant au même modèle recherché. En effet, un modèle correspond à la trace d'un évènement dans une série et deux évènements identiques ne peuvent pas se recouvrir.

Cependant, lorsqu'on recherche des segments de natures différentes, on ne peut pas affirmer *a priori* qu'ils ne peuvent pas se superposer. C'est à ce niveau que les agents interviennent : la segmentation est réalisée par plusieurs *agents de segmentation*. Dans notre méthode de segmentation, la zone de masquage n'est bien utilisée que par un *agent de segmentation* uniquement pour lui-même. Les autres agents ne perçoivent pas la zone de masquage. Ainsi les segments produits par deux *agents de segmentation* distincts peuvent se superposer.

C.2 Construction de modèles d'évènements

On présente tout d'abord la méthode générale qui permet de classer les segments récupérés de la segmentation et de générer les modèles de ces évènements. Nous présentons ensuite l'algorithme de classification qui permet au système de construire les classes de segments en prenant en compte les perturbations liées aux interactions de l'utilisateur.

C.2.1 Vue générale de la construction de modèles d'évènements

La construction des modèles d'évènement est la tâche d'un *agent de construction de modèles* qui récupère l'ensemble des segments d'un type de séries auprès des *agents de segmentation*.

Le processus général suivi par un agent est illustré par la figure 3.18. Les segments sont tout d'abord transformés sous une forme vectorielle (Étape (1)). On a vu dans la section B.2.1 que les distances entre segments étaient problématiques et que nous utilisons donc des modèles de segments sous forme vectorielle pour s'affranchir de cette difficulté. Les similarités entre segments établies par l'utilisateur sont traduites par des liens entre les exemples (représentés en rouge sur la figure). Les exemples sont ensuite traités par

un algorithme de classification non supervisée⁷ (Étape (2)). L'objectif est alors de construire des classes de segments qui soient *homogènes*. Chaque classe de segments est alors représentative d'un évènement. Dans la dernière étape (Étape (3)), on utilise un algorithme d'extraction des caractéristiques des segments d'une classe pour en construire son modèle (modèle d'évènement).

C.2.2 Introduction à l'algorithme de classification

Les algorithmes de classification non supervisée tels que les K-Means se basent sur une métrique adaptée aux données traitées. Lorsqu'un utilisateur dispose d'une connaissance *a priori* sur le critère d'homogénéité des classes, il doit alors construire une métrique qui correspondra à son attente. Généralement, cette métrique est construite de manière non intuitive. Certaines méthodes [226] permettent de s'appuyer sur les compétences naturelles de l'utilisateur (*e.g.* sa capacité à décider de la similarité de deux exemples) pour l'aider à définir cette métrique.

Cependant, nous sommes dans la situation où la connaissance d'un utilisateur n'est pas suffisamment formalisée ni pour classer des exemples ni pour donner de critères de similarité/dissimilarité pour comparer les exemples. Nous avons donc proposé comme modèle de segment un modèle dans lequel les distances sont adaptées 1) de sorte à extraire la métrique la plus adaptée pour l'interprétation et 2) de façon indépendante pour chaque évènement (*i.e.* pour chaque classe de segment).

Par conséquent, l'algorithme de classification doit à partir d'un ensemble d'exemples et en même temps⁸ :

- construire les classes qui disposent chacune d'une distance propre et caractéristique. Le nombre de classe est également à trouver.
- construire les distances caractéristiques des classes.
- faire l'attribution des exemples aux classes les plus adaptées.

Pour cela, il est dirigé par l'objectif de construire une classification dans laquelle tous les éléments d'une classe sont *similaires* entre eux et *dissimilaires* aux éléments des autres classes. La notion de similarité n'étant pas partagée par toutes les classes, les critères de *similarité/dissimilarité* sont propres à chaque classe.

On attend également de l'algorithme qu'il prenne en compte les liens entre les exemples de sorte que deux exemples liés soient associés à une même classe.

C.2.3 Algorithme de clustering/classification

Soient m exemples, vecteurs de dimensions n . On appellera "caractéristique" l'une des composantes des vecteurs exemples. Les éléments du vecteur sont compris entre 0 et 1. La normalisation sert à assurer qu'il n'y aura aucune prévalence d'une caractéristique sur une autre (*cf.* paragraphe B.2.3).

Dans la suite de cette section, on introduit tout d'abord nos propres définitions de classes et de distances qui permettent ensuite de présenter l'algorithme.

Les classes Une classe est constituée d'un ensemble d'exemples et également d'une distance qui lui est propre. Cette distance permet de comparer un exemple de la classe avec un autre exemple (de la classe ou extérieur à cette classe). En particulier, tous les éléments de la classe entre eux.

La spécificité de l'algorithme est bien dans l'attribution à chaque classe d'une distance particulière qui est en fait caractéristique de celle-ci. On instaure ainsi un fort couplage entre les exemples qui sont attribués à la classe et la distance de celle-ci. D'une part, les exemples de la classe doivent être suffisamment⁹ proches entre eux, selon la distance de la classe et d'autre part la distance doit être représentative des exemples qui constituent la classe. En conséquence, l'équilibre est atteint lorsque la distance est bien adaptée aux exemples et que les exemples qui sont bien adaptés à la distance¹⁰.

⁷ On distingue les algorithmes de classification non supervisés (ou algorithmes de clustering) des algorithmes de classification. Les premiers disposent d'un ensemble d'exemples à partir desquels ils construisent des groupes homogènes (les classes) alors que les seconds disposent déjà des classes, ils réalisent l'attribution de la *bonne* classe à chaque exemple.

⁸ Cette classification n'a pas de prétention à afficher de meilleurs résultats que les algorithmes de classification qui s'appuient sur des résultats de statistiques dûment prouvés et éprouvés.

⁹ "Suffisamment proche" signifie techniquement "en dessous d'un seuil fixé" *a priori*.

¹⁰ On retrouve ici la complémentarité du global et du local qui déjà été vu lors de la présentation de l'interprétation. Le parcours interprétatif qui permet d'obtenir une cohérence (l'état d'équilibre de la classification) est le chemin qui permet d'arriver à la classification des exemples.

Ensemble de segments :

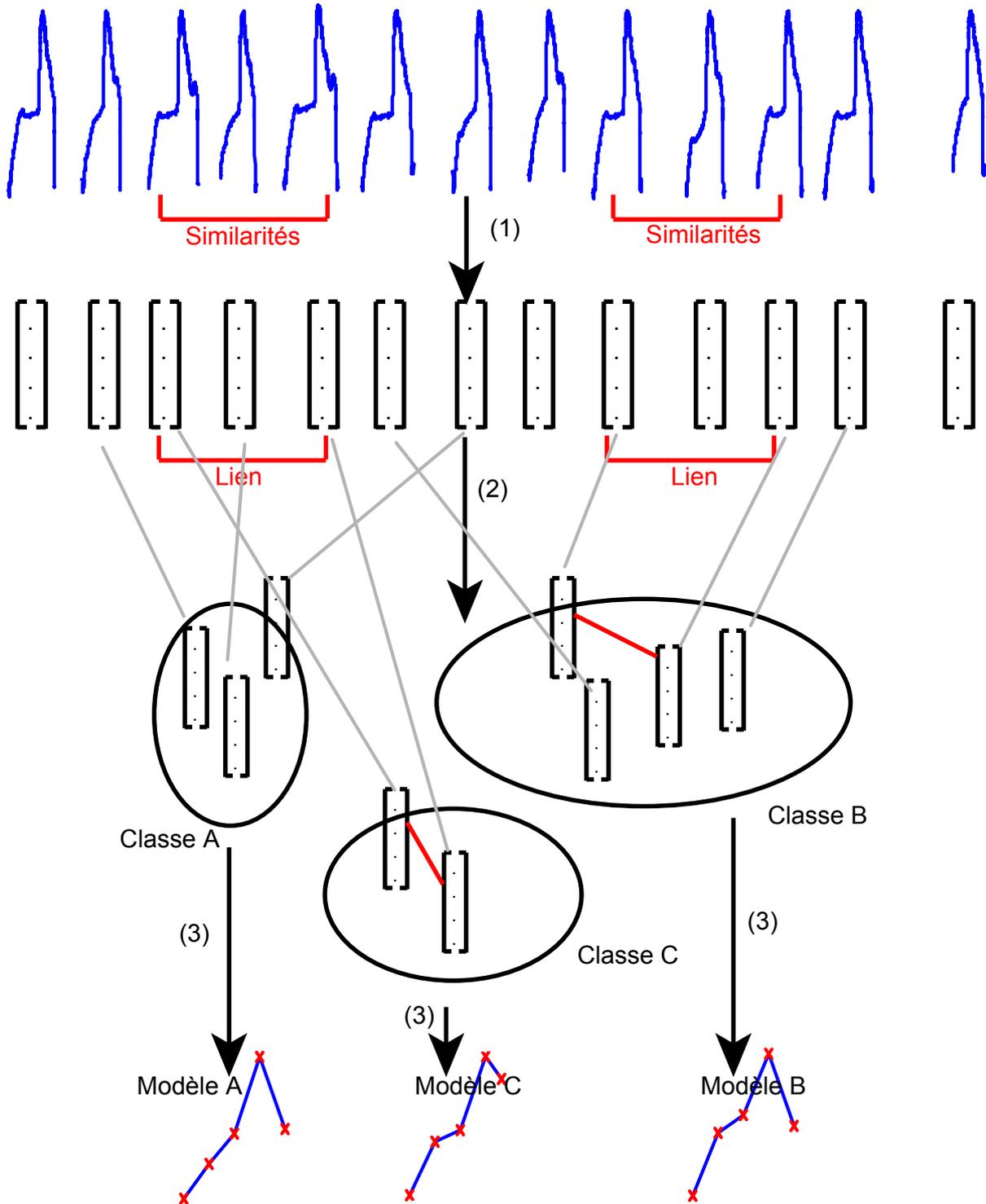


FIG. 3.18. Illustration de la méthode de classification des segments pour la constitution des classes de segments et des modèles d'évènements.

Les distances Pour nos données vectorielles provenant de la modélisation des segments, la distance entre vecteurs à classer est une distance Euclidienne pondérée :

$$d_C(X, Y) = \sqrt{\sum_{i=1}^n \alpha_i^C (x_i - y_i)^2}$$

Les α_i^C , qui constituent un vecteur α de dimension n , sont ici les pondérations utilisées pour le calcul de la distance d'un élément de la classe C à un autre élément. Le vecteur α caractérise donc la distance propre à C . C'est en adaptant ce vecteur qu'on adapte la distance de la classe.

Un algorithme de *feature selection* telle qu'une analyse en composantes principales peut extraire les caractéristiques principales des exemples de la classe et fournir le vecteur α dont les composantes seront faibles lorsqu'il y aura une forte disparité entre les valeurs de cette composante dans les exemples, ou au contraire une forte composante si les valeurs sont très similaires. Ainsi, la distance est bien caractéristique des exemples et permet d'avoir des mesures entre exemples faibles, calculées par la distance Euclidienne pondérée avec α .

Ce vecteur α réifie donc la caractérisation de la classe par la distance. En conséquence, la diagonale de la matrice distance d'un modèle d'évènement (notée D^E dans la section B.3) est définie par les valeurs de ce vecteur¹¹.

La distance étant caractéristique d'une classe, elle sert également à comparer les classes entre elles. Ceci permettra en particulier d'ajuster le nombre de classes. Lorsque deux classes sont trop *proches*, c'est que certainement, il n'y avait pas de raison de les distinguer, on pourra alors regrouper tous les exemples dans une seule et même classe avec une distance adéquate.

On exprime la distance qui existe entre deux classes par une méta-distance calculée à partir des vecteurs α . Soient les classes C^1 et C^2 qui ont donc chacune leur distance caractérisée respectivement par les vecteurs α^{C^1} et α^{C^2} . La distance entre deux classes est calculée par la distance Euclidienne entre les vecteurs α :

$$md(C^1, C^2) = \sqrt{\sum_{i=1}^n (\alpha_i^{C^1} - \alpha_i^{C^2})^2}$$

Survol pratique de l'algorithme On s'appuie sur la figure 3.19 pour présenter l'algorithme. Sur la gauche de la figure se trouve l'ensemble des exemples qui restent à classer, et sur la droite, les *patatoïdes* représentent les différentes classes qui ont été construites avec leurs exemples¹².

Dans notre classification, les exemples sont ajoutés itérativement à la classification en cours de construction. Et la classification est raffinée à chaque introduction d'un nouvel exemple. Outre les étapes d'adaptation qui permettent en particulier d'adapter le nombre de classes, l'algorithme s'apparente beaucoup avec un algorithme de *K-Means*. L'algorithme comporte ainsi sept étapes :

1. Tirer un exemple e au hasard dans la liste des exemples à classer.
2. Parmi les classes existantes, chercher la classe C qui convient le mieux à e . Au besoin, une nouvelle classe C est créé.
3. Intégrer e à C
4. Adapter C , *i.e.* ajuster la distance aux données.
5. Vérifier la *consistance* de C
 - Si un exemple n'est pas consistant, on l'éjecte et le replace dans la liste des exemples à classer et on repasse à 4),
 - Sinon, on passe à 6).
6. Vérification des *typicités des classes*.

¹¹ Le choix de l'utilisation d'un vecteur pour réaliser la distance entre classes et non pas une matrice qui permettrait de disposer d'une distance de Mahalanobis est lié à notre méthode d'extraction de caractéristiques d'une classe (*cf.* Annexe B) qui rend difficile l'adaptation d'une matrice.

¹² En fait, cette visualisation est figurative car l'utilisation de distances propres à chaque classe est un obstacle à la visualisation de la classification dans son ensemble

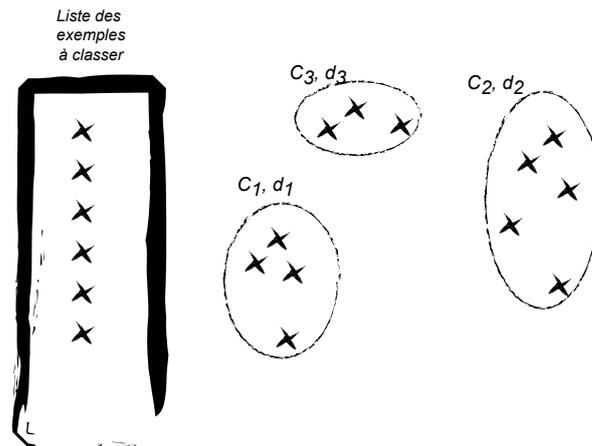


FIG. 3.19. Illustration des éléments de la classification.

7. S'il reste des exemples dans la liste des exemples, revenir à (1).

L'état d'équilibre à atteindre est défini par deux critères :

- *Consistance des classes* (i.e. forte similarité intra-classe) : Tous les distances entre les exemples d'une même classe doivent être inférieures à un seuil s .
- *Typicités des classes* (i.e. forte dissimilarité inter-classes) : Toutes les distances entre les classes doivent être supérieures à un seuil fT .

Pour avoir plus de précisions sur les méthodes qui sont utilisées (Adaptation d'une distance, éjection d'un exemple, vérification de la consistance) et les paramètres importants de l'algorithme, on pourra consulter l'Annexe B.

En particulier, certaines solutions pratiques ont été ajoutées à l'algorithme décrit pour éviter des comportements non désirables et envisageables du système. Par exemple, on peut noter qu'il peut exister des problèmes de bouclages qui posent le problème de la convergence. Le bouclage le plus simple (ordre 1) qui peut apparaître étant un exemple qui peut intégrer une classe, mais une fois celle-ci adaptée, il en est éjecté. La classe retrouve alors ses caractéristiques initiales et l'exemple peut alors de nouveau être intégré dans la classe. Mais des bouclages d'ordres supérieurs peuvent apparaître. La solution consiste à dire qu'au bout d'un certain nombre de tentatives d'amélioration de la classification, les exemples qui ne sont pas classés restent ainsi.

Prise en compte des interactions de l'utilisateur Les perturbations introduites par les interactions avec l'utilisateur s'intègrent directement dans l'algorithme de classification.

Les modifications d'attribution d'un segment à un évènement correspond à un déplacement d'un exemple dans la classe correspondante. La modification engendre alors une nouvelle adaptation des classes modifiées, l'algorithme poursuit son cours pour se stabiliser de nouveau. Comme il avait été indiqué dans la présentation des interactions (cf. section A.3.3), la conséquence d'une perturbation n'assure pas de conserver l'exemple introduit dans la classe initiale et une seule perturbation peut engendrer de nombreuses modifications globales de classification.

Les liens sont également intégrés dans l'algorithme. Pour cela, on force les exemples à toujours appartenir à une même classe. Le déplacement d'un exemple dans une classe conduit tous les exemples qui y sont liés à intégrer la même classe. Récursivement, les exemples liés à des exemples se retrouvent ainsi tous dans la même classe. L'adaptation de la classe se fait alors en prenant en compte l'ensemble des nouveaux exemples qui ont été introduits. Cette méthode assure ainsi que deux segments liés seront toujours associés à un même évènement mais il permet en même temps à l'algorithme de générer ses propres classes, i.e. ses propres représentations des exemples.

C.2.4 Classification comme système autopoïétique, adaptatif

Cet algorithme est une implémentation d'un processus d'"auto-organisation" des classes et des exemples qui effectue des itérations jusqu'à la stabilisation.

L'utilisation de cet algorithme répond aux contraintes théoriques que nous nous sommes imposées pour concevoir un système inspiré des concepts proposés par Maturana et Varela. La classification présentée est *en quelque sorte* un système autopoïétique, adaptatif. L'objectif de cette section est de mettre en évidence certains rapprochements entre l'inspiration des concepts et la solution algorithmique proposée, mais il n'est pas d'attribuer au système les propriétés cognitives riches des systèmes vivants étudiés par Varela et Maturana.

À la base, la classification est constituée d'exemples et les classes constituent une structure qui organise les exemples. Le processus mis en place permet de construire une structure et d'en assurer une continuité et ce malgré les perturbations extérieures (ajouts, suppressions et déplacements d'exemples). Avec ces règles de construction mises en place, le système construit donc sa propre structure. Ceci permet donc de l'assimiler à un processus autopoïétique. La plasticité de la méthode qui a été mise en évidence par sa capacité à intégrer les interactions de l'utilisateur est une conséquence de ce caractère autopoïétique.

De plus, cette classification s'apparente à un processus expérientiel puisque l'ordre d'apparition des exemples dans la classification et l'ordre des modifications influent sur le résultat de la classification¹³.

Même les contraintes pratiques de bouclages et peuvent être vues comme le caractère non téléologique de notre classification. Et la solution pour y remédier est la solution qui est souvent faite par des interpréteurs humains qui peuvent mettre de côté certains exemples qui ne leur semblent pas pertinents de prendre en compte dans leur interprétation au risque de perturber une consistance (satisfaisante) déjà acquise.

Le rapprochement de cet algorithme avec le paradigme qui nous a inspirés le système dans son ensemble fait qu'il est un élément central dans la dynamique globale du système. Dans sa conception actuelle, une grande partie de la dynamique du SMA est permise grâce à cette classification et ses propriétés particulières.

C.3 Apprentissage de scénarios

L'apprentissage de scénarios doit permettre de construire des scénarios à partir des séries temporelles symboliques qui ont été recueillies de la symbolisation des séries temporelles. Le scénario doit être significatif soit parce qu'il représente des co-occurrences d'événements fréquentes, soit parce qu'il intéresse un clinicien qui cherche à comprendre l'occurrence d'événements particuliers (afin d'identifier des scènes rares mais intéressantes en pratique).

Nous présentons d'abord le mécanisme de construction d'exemples de scénarios qui permet de superviser la construction d'une explication. Nous présentons ensuite différentes approches de construction d'un scénarios à partir d'exemples et présentons en plus détaillé celui qui a été utilisé pour nos *agents de construction de scénarios*.

C.3.1 Construction d'exemples pour une explication

La notion d'explication d'un événement E permet de construire des exemples du scénario qui va expliquer les occurrences de E . La construction des exemples d'une explication d'un événement E se fait en trois étapes :

1. On repère toutes les occurrences de E . Il y aura autant d'exemples que l'occurrence de E . Par conséquent, l'apprentissage d'un événement rare pourra se faire de la même manière que pour les événements fréquents. On ne sera pas en mesure de construire des scénarios aussi robustes.
2. On construit un exemple en concaténant l'ensemble des symboles qui se trouvent dans une fenêtre temporelle précédent chaque occurrence de E . L'ensemble des symboles qui constituent un exemple dépend de la taille de la fenêtre. Plus la taille de fenêtre est grande plus on considère une influence grande des événements dans le temps. La taille de fenêtre doit être définie *a priori*. Mais une

¹³ En général, ce comportement est plutôt à proscrire pour un algorithme de classification, c'est en particulier pourquoi il n'a pas prétention à rentrer en compétition avec d'autres algorithmes de classifications.

connaissance de la dynamique générale du système d'où proviennent les symboles facilite le choix de cette valeur.

3. On aligne les dates des événements sur la date de E . L'alignement des dates permet d'avoir une base de référence temporelle commune pour tous les exemples. Comme dans une explication on construit des relations temporelles entre les symboles du motif explicatif et le symbole à expliquer, l'apprentissage nécessite d'avoir les durées entre les symboles d'un motif et le symbole à expliquer et non des dates absolues. C'est ce que permet d'obtenir l'alignement.

On constitue ainsi une base d'exemples à partir de laquelle on va chercher à construire un (des) scénario(s) pouvant expliquer E . Les dates des exemples étant alignées, l'algorithme d'apprentissage doit donc se focaliser sur la construction des similarités dans les exemples en prenant en compte la dimension temporelle.

L'utilisation de la notion d'explication est un avantage important par rapport à une recherche de scénarios dans l'ensemble des séries symboliques. En effet, la recherche exhaustive des scénarios est très difficile car on doit utiliser la globalité des séries temporelles. En s'intéressant à la construction d'explications, on peut focaliser le traitement sur des parties des séries temporelles.

En ayant une base d'exemples on va pouvoir d'une part utiliser un algorithme d'apprentissage supervisé. En revanche, l'algorithme d'apprentissage de chroniques de Dousson et *al.* [83] réalise un apprentissage non supervisé dans l'ensemble des séries symboliques. Et d'autre part, on peut aussi bien réaliser un apprentissage de scénarios fréquents mais également des explications pour des événements rares. Les approches usuelles de construction de scénarios ne considèrent pas ce cas.

C.3.2 Choix d'une méthode parmi les méthodes existantes

On présente tout d'abord un court état de l'art non exhaustif des méthodes pouvant être utilisées pour l'apprentissage de scénarios ou d'objet s'en approchant. On introduit simplement quelques formalismes de représentation des scénarios et on leur associe des algorithmes qui permettent de les traiter en donnant quelques informations sur les limites de ceux-ci. On revient ensuite plus en profondeur sur une extension de l'algorithme APriori que nous utilisons pour nos *agents de construction de scénarios*.

Cet état de l'art permet finalement de réaliser un choix basé sur notre besoin d'apprendre des contraintes temporelles quantitatives.

Méthodes logiques Les exemples de scénarios peuvent être décrits par des ensembles de prédicats logiques. L'apprentissage de scénarios consiste alors à appliquer les méthodes génériques d'induction, par exemple l'*Inductive Logic Programming* de DeRaedt et *al.* [29], qui vont extraire les similarités entre les exemples.

Si des logiques de description très expressives permettent de représenter toute la dimension temporelle (quantitative et qualitative) des scénarios, elles ne sont pas utilisables par les méthodes d'induction. La logique de McDermott [150] est une logique de points qui permet de conserver la structure linéaire du temps. Les prédicats peuvent être associés à des dates numériques. C'est une logique de description très expressive, mais en pratique les méthodes d'induction logique sur cette logique ne sont pas applicables. Au mieux elle permet de faire de la vérification de satisfaction de contraintes. On retrouve les mêmes difficultés dans l'utilisation de logiques modales.

Les méthodes d'induction fonctionnent avec des logiques propositionnelles. Il faut alors pouvoir représenter le temps à l'aide de propositions. La méthode plus connue est certainement la logique d'Allen [6] qui est une logique d'intervalles qui permet d'utiliser les sept prédicats décrivant les relations temporelles qualitatives élémentaires (Égal, avant, pendant, rencontre, recouvre, commence, fini). Dans le système de Quiniou et *al.* [173], l'abstraction du temps est faite sous la forme de prédicats (**court**, **normal**, **long**). Grâce à l'introduction de connaissance *a priori* (*Background Knowledge*) l'ILP permet d'apprendre l'intervalle des valeurs définissant les prédicats **court**, **normal** et **long**. Mais dans les deux cas, il a fallu opérer à une réduction drastique de l'information temporelle.

Les logiques floues permettent de prendre en compte des attributs dates sur lesquels on peut apprendre des distributions, le problème est alors de proposer des algorithmes qui soient capables d'apprendre à la fois les enchaînements de symboles et des distributions.

Dans la mesure où nous souhaitons prendre pleinement en compte la dimension temporelle quantitative

dans les scénarios, nous n'avons pas retenue cette solution.

Méthodes à base de graphes Le formalisme des graphes, par leurs dimensions symbolique et temporelle (transitions orientés et pondérables) permet de représenter les scénarios : les graphes temporisés ([82]), les réseaux Bayésiens associés aux modèles de Markov cachés ([35]) ou encore les réseaux de Petri temporisés ([43]). Il peut alors également être fait une différence entre la représentation du temps par dates ou intervalles. Les réseaux Bayésiens sont un moyen de représenter des séquences d'évènements à temps discrets. Si on veut exprimer des durées d'évènements, on pourra aisément échantillonner à l'aide de la plus petite durée d'évènement et au besoin répéter un évènement pour exprimer la durée. L'expressivité est alors limitée par l'impossibilité de différencier "deux fois l'évènement A de durée t " et l'occurrence de "l'évènement A de durée de $2t$ ". Les réseaux de Petri temporisés ou les graphes temporisés permettent de décrire de façon plus fine des transitions entre évènements en spécifiant la durée qu'il doit y avoir entre chacun d'entre eux ce qui permet de résoudre le problème précédent.

Pour l'apprentissage, on retrouve le même type de problème que pour les logiques floues. L'apprentissage de la structure du réseaux *et* des contraintes temporelles n'est pas évident en toute généralité. Pour les réseaux Bayésiens, la structure est généralement donnée et les distributions des transitions sont apprises à l'aide du formalisme en HMM. Dans [43], les auteurs présentent des méthodes d'apprentissage pour lesquelles ils limitent l'apprentissage de la dimension symbolique du motif. En partant d'un pré-motif, l'algorithme permet d'apprendre les durées des transitions et de supprimer également certaines transitions et états qui ne sont pas significatifs.

Méthodes à base de séquences La communauté des bases de données développe des méthodes intéressantes pour l'exploration d'*itemsets*, c'est-à-dire des ensembles d'*item*. Un *itemset* est un objet assez proche du motif d'un scénario. Les méthodes de construction d'*itemsets* peuvent servir de base à la proposition de méthode de construction de scénarios.

En premier lieu, ces méthodes ont été étendues pour la construction de séquence, c'est-à-dire des *itemsets* dans lequel l'ordre des *items* est important (motif ordonné). Des revues des méthodes de recherche de séquences ont récemment été dressées par Massegla [146] ou Zhao [227].

D'autres travaux s'attachent non seulement à construire des séquences mais à également considérer quantitativement le temps dans les motifs. Des approches comme celles proposées par Moskovitch [157] ou Duchêne *et al.* [88] utilisent la répétition de symboles ayant des durées atomiques fixées.

Duchêne *et al.* [88] utilisent une méthode de projections aléatoires de séquences illustrée sur la figure 3.20.

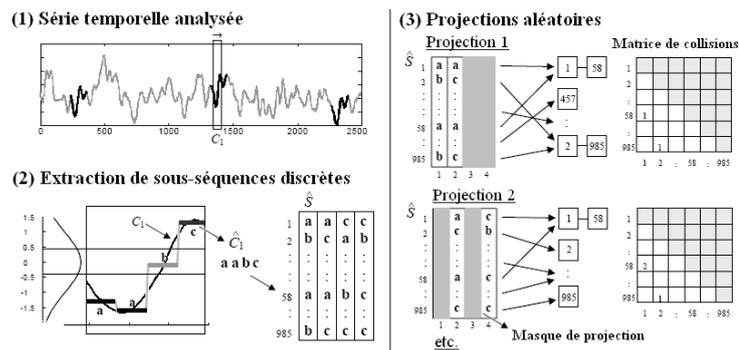


FIG. 3.20. Méthode de construction de motifs fréquents par Duchêne *et al.* (Figure extraite de [87]).

Pour deux raisons, nous ne nous orientons pas vers des méthodes qui utilisent la répétition de symboles pour prendre en compte des durées "quantitatives" :

- Pour des raisons calculatoires : les algorithmes de construction de séquences sont très souvent de complexité exponentielle avec la taille des motifs construits. Ne sachant pas définir *a priori* la granularité des durées des évènements, nous ne pouvons pas prendre le risque de devoir construire des motifs comportant un grand nombre de symboles (*i.e.* >10).

- Pour des raisons de sémantique : un symbole dans la sémantique que nous nous sommes fixée au premier chapitre est associé à un évènement physiologique qui à un début et une fin. Par conséquent, deux symboles successifs correspondent à la répétition de cet évènement et non à un unique évènement qui a duré deux fois son temps atomique.

Nous préférons donc nous intéresser à des méthodes qui permettent la construction de relations quantitatives codées comme des durées. La plus utilisée des approches à base de séquences adaptée pour l'apprentissage de scénarios, *i.e.* avec des relations quantitatives (numériques) et/ou qualitatives, est certainement l'algorithme APriori.

Adaptation de l'algorithme APriori Nous détaillons un peu plus cette approche et en particulier l'algorithme de Dousson [83], car c'est l'algorithme qui est actuellement implémenté dans nos *agents d'apprentissage de scénarios*.

La méthode APriori permet à Agrawal et Srikant [3] de construire les règles d'associations les plus grandes et de fréquence minimum fixée (f). Cet algorithme a été alors adapté pour la construction de séquences fréquentes les plus grandes à partir de séquences exemples. La fréquence qui est calculée donne la *confiance* du système dans la séquence construite. Le principe étant toujours fondé sur le théorème suivant dont la preuve est triviale par l'absurde :

Théorème : Un motif de dimension $N + 1$ ne peut être de fréquence supérieure à f si tous les sous-motifs qui le composent (en particulier ceux de dimension N) ne sont pas au moins de fréquence f .

L'algorithme APriori (*cf.* figure 3.21) répète l'exécution de deux étapes sur des motifs de plus en plus grands :

1. une étape d'estimation des candidats de taille n en cours de construction afin d'élaguer ceux qui ne sont pas de fréquence suffisamment grande. Le théorème nous indique que si un motif n'est pas suffisamment fréquent, alors tout les motifs le contenant ne le seront pas non plus. Donc cela ne sert à rien de le considérer par la suite.
2. une étape de construction de candidat de taille $n + 1$ à partir des motifs de taille n . À cette étape on doit vérifier que tous les sous-motifs sont fréquents.

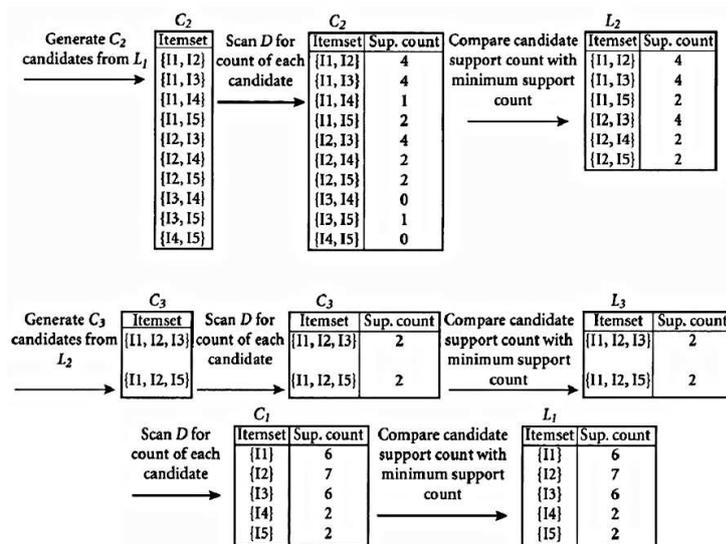


FIG. 3.21. Illustration de l'algorithme APriori sur la construction des plus grands *itemsets* fréquents.

Ces deux étapes sont dualement complexes. Si initialement la tâche d'estimation de la fréquence est calculatoire car il y en a de nombreux possibles, la génération de nouveau motifs est simple car peu de contraintes sont à vérifier. Mais au fur et à mesure que les motifs grandissent, il est plus compliqué de

générer de nouveaux motifs que d'en estimer la fréquence. En pratique, il existe différentes heuristiques qui permettent de parcourir les motifs possibles de manière incomplète mais plus efficace.

La notion d'*itemset* est assez proche de la notion d'explication, puisqu'il suffit d'ajouter à chaque *item* une date et une durée. Cependant, si on dispose d'un algorithme qui est capable d'extraire la structure du motif, la difficulté reste toujours l'apprentissage des contraintes temporelles. Plusieurs proposition d'adaptation de l'algorithme APriori.

Nous proposons ici d'utiliser l'algorithme de C. Dousson [83], adaptation temporelles de l'algorithme APriori. D'autres variantes de l'algorithmes existent par exemple une proposition de Mannila et *al.* [141], Papapetrou et *al.* [167] ou Kam et Fu [126]. Nous avons nous même proposé une variante de cet algorithme [105] que nous donnons en annexe (*cf.* Annexe C). La méthode de Hoppner [117] découle également de l'algorithme APriori avec un apprentissage des contraintes temporelles sous la forme de relations d'Allen.

Le système FACE de C. Dousson [83] permet de construire des "chroniques" qui sont des scénarios avec des contraintes temporelles qui sont quantitatives, (Chaque contrainte est représentée par un intervalle des durées acceptables entre deux événements) et entre des événements instantanés (*i.e.* les événements ne sont pas associés à des durées).

Une extension du théorème de base de l'algorithme APriori permet d'appliquer la même stratégie de construction de motifs de plus en plus grand en élaguant les branches qui ne conduiront *a priori* uniquement à de mauvais candidats. Pour cela, la notion de sous-chronique d'une chronique \mathcal{C} est une chronique à laquelle un *item* a été supprimé de \mathcal{C} et pour lequel tous les intervalles définissant les contraintes temporelles contiennent les intervalles de \mathcal{C} correspondants (*i.e.* ceux entre les mêmes *items*).

L'algorithme contient maintenant trois étapes :

1. Une étape de génération des candidats non contraints qui permet de construire tous les motifs possibles de taille supérieure sans tenir compte des contraintes temporelles ;
2. Une étape de vérification de la plausibilité des candidats avec les contraintes temporelles : c'est-à-dire qu'en plus d'être des motifs avec des sous-motifs fréquents, il faut que les contraintes temporelles soient consistantes entre elles. À cette étape, les intervalles définissant les contraintes sont éventuellement élargis pour rendre les motifs consistants ;
3. Dans une troisième étape d'estimation de la fréquence d'apparition de ces motifs et d'élagage des motifs non fréquents.

On peut noter que dans cet algorithme les durées entre les *items* ne sont pas vraiment appris. Ils sont obtenus uniquement comme le résultat de la production de chroniques plus générales et consistantes. Les contraintes obtenues sont au final les plus larges des scènes qui sont représentées.

Nous avons adapté cet algorithme pour non seulement extraire les durées des transitions entre les événements, mais également la durée des événements eux-mêmes. La structure d'explication (transitions uniquement entre un symbole de l'explication et le symbole à expliquer) fait que l'étape de vérification de contraintes temporelles est très simple. Il suffit d'extraire les durées d'événements qui sont les plus grandes.

C.4 Reconnaissance de scénarios

Pour un *agent de reconnaissance de scénarios*, deux situations peuvent être rencontrées :

- s'il reconnaît une scène qui correspond au scénario, alors l'agent annote l'enregistrement avec une scène correspondant à ce scénario.
- si la reconnaissance n'est que partielle (*i.e.* que sur l'ensemble des symboles qui constituent un motif, l'un d'entre eux n'a pas été trouvé), cela nous donne l'opportunité d'améliorer la symbolisation.

La reconnaissance d'un scénario est réalisée de manière similaire à l'apprentissage. L'agent en charge de reconnaître un scénario parcourt l'ensemble des enregistrements à la recherche des occurrences des événements qu'explique son scénario. Lorsque c'est le cas, il observe la correspondance entre les symboles précédents l'occurrence et le scénario appris en appliquant une projection comme indiqué par la figure 3.15. En dessous d'un seuil fixé *a priori*, on vérifie que l'ensemble des symboles est présent. Cette seconde vérification sert notamment à éviter de systématiquement ne pas considérer les symboles de petite durée, mais elle nous est également indispensable pour attribuer les symboles à la scène en cours de construction.

Si ces deux conditions sont remplies, alors un scénario a été repéré et l'agent peut ancrer l'annotation d'une scène.

Si la seconde vérification n'est pas avérée (*i.e.* si un symbole ne correspond pas, au delà d'un symbole, la scène n'est pas reconnue), on peut faire l'hypothèse qu'il peut y avoir eu une erreur dans les étapes précédentes (segmentation ou symbolisation). L'*agent de reconnaissance* réalise l'annotation avec la scène et propose alors une modification de la série symbolique en introduisant un symbole correspondant à son attente.

Il s'agit d'une proposition dans le sens où le système va essayer de corriger une erreur sans être sûr de la pertinence de la modification. Seule la suite du processus global d'interprétation permettra de juger de l'intérêt de l'insertion d'un tel symbole : s'il est conservé par le système l'insertion aura été utile puisqu'on augmentera la confiance du scénario correspondant (une occurrence de plus lors du prochain apprentissage), et s'il n'est pas conservé alors la modification n'aura pas été fructueuse.

De nouveau, avec l'introduction d'un *feedback*, le spectre du bouclage guette le processus. Si la symbolisation n'"accepte" pas la modification et que cela conduit à proposer les mêmes exemples pour l'apprentissage, on boucle avec la même demande de modification.

C.5 Motivation des choix des algorithmes

Nous concluons ce chapitre en précisant les motivations de nos choix des algorithmes. Ces motivations sont de trois types :

- Répondre spécifiquement à la caractéristique d'adaptation de la perception et des traitements demandés par l'approche Enactive. En particulier, c'est à travers le caractère plastique du système que peuvent être pris en compte les modifications de l'utilisateur. Donc les algorithmes ont été choisis également pour la possibilité de s'adapter au contact des interactions de l'utilisateur au travers des annotations.
- Faire une interprétation riche, c'est-à-dire que les annotations et en particulier les modèles de l'interprétation réalisée par le système doivent potentiellement être sémantiquement riches pour identifier dans les séries temporelles des événements et des scénarios informatifs. Nous espérons ainsi d'une part couvrir les besoins d'expression du contexte applicatif et également découvrir de nouveaux modèles.
- Décentraliser les traitements et utiliser le paradigme multi-agents afin de faciliter l'exploration de données massives, disposer d'une dynamique des traitements facilitant l'émergence et dans la perspective d'extension à d'autres algorithmes.

Le choix de l'algorithme de segmentation est motivé par les trois types. Tout d'abord, l'utilisation d'un algorithme à base de fenêtres flottantes permettait de segmenter une série temporelle sur des critères sémantiquement riches puisqu'il permettait de le faire à partir de modèles expressifs, ensuite, il était facile de distribuer le traitement. Mais ce qui motive le plus le choix de cette méthode est la possibilité de faire une segmentation à partir des modèles d'événements qui sont construits par le système lui-même. Or, dans la mesure où la segmentation est l'*organe de perception* des séries temporelles de notre système, du point de vue Enactif, il offre la possibilité d'avoir une adaptation de la perception des séries temporelles au cours de l'interprétation.

Pour l'algorithme de classification, on ne peut pas vraiment parlé de choix, car il s'agit plutôt d'une proposition spécifique à notre approche qui offre la possibilité au système sa capacité à construire la dimension paradigmatique du sens en suivant les principes de l'Enaction. L'algorithme propose des classes qui ne sont pas fondés sur une représentation et sur la seule consistance de la classification. De plus, cette consistance ne s'évalue que grâce aux distances qui sont elles-mêmes propres au système et à l'historique de l'interprétation.

Finalement, le choix des algorithmes d'apprentissage et de reconnaissance de scénarios sont principalement motivés par l'expressivité qu'ils apportent dans la dimension syntagmatique du sens qui est construit par le système. En effet, la prise en compte de la temporalité de l'apprentissage est un supplément d'expressivité qui a pleinement son intérêt lorsqu'on s'intéresse à des dynamiques des paramètres dont on interprète la trace. L'introduction de la notion d'explication a été motivé par la possibilité qu'elle offre pour décentraliser les traitements.

Chapitre 4

Évaluations

L'évaluation de notre travail doit permettre d'évaluer le système en tant que système collaboratif. Nous utilisons pour cela une méthodologie en trois étapes : l'évaluation de la faisabilité de l'approche collaborative, de la pertinence du couplage et de l'utilisabilité du système.

Pour l'évaluation de la faisabilité, nous présentons d'abord les résultats des expérimentations des solutions techniques de traitements des données puis des résultats sur le bouclage du système. Ensuite, nous présentons des résultats de pertinence du couplage obtenus par l'expérimentation de notre système par des utilisateurs, et enfin une utilisation de notre système sur des données réelles de patients ventilés.

Pour réaliser ces expérimentations, nous présentons le prototype CATS (Collaboration Annotation of Time Series) qui est une implémentation du système multi-agents proposé.

A Méthodologie d'évaluation de l'interprétation collaborative

L'évaluation de ce travail a pour objectif de permettre de discuter l'approche Enactive de la collaboration pour l'interprétation de séries temporelles. Pour mener à bien cette évaluation, nous nous sommes appuyés sur un cadre méthodologie de l'évaluation d'une approche collaborative que nous présentons dans la section suivante. Et afin de mener à bien les évaluations proposées dans ce cadre, nous avons développé un prototype (nommé CATS) qui a été appliqué à des données réelles et simulées que nous présentons également dans une dernière section de la présentation de la méthodologie.

Dans ce chapitre, nous ne donnons que les résultats de nos expérimentations et les interprétations immédiates qui peuvent en être faites. Dans le chapitre suivant, nous dressons des conclusions plus globales sur notre travail.

A.1 Cadre méthodologique des évaluations

L'évaluation d'une collaboration homme-machine est une tâche difficile. Il n'existe pas vraiment de méthodologie d'évaluation des systèmes collaboratifs. Plus précisément, il en existe beaucoup [125] et aucune se dégage comme un standard auquel il nous semble devoir adhérer.

Nous proposons un cadre méthodologique qui s'inspire de la méthodologie de Shyr et *al.* [193]. Cette méthodologie est utilisée pour l'évaluation de systèmes à initiative mixte. Elle organise les expérimentations à mener selon trois dimensions : (1) les qualités du système qui sont évaluées (la performance et l'utilisabilité), (2) le contexte dans lequel sont réalisées les évaluations (s'agit-il d'évaluations en contexte réel ou non ?) et (3) les types de critères utilisés (objectifs ou subjectifs).

Nous adaptons l'organisation des évaluations selon les qualités (la performance et l'utilisabilité) attendues du système. À la place d'une évaluation de la performance du système, nous parlons plutôt d'une évaluation de la **pertinence de l'approche Enactive** pour réaliser la collaboration. Cette évaluation permet de prendre le point de vue des utilisateurs sur l'approche proposée. L'évaluation de l'utilisabilité du système devient simplement l'évaluation de l'**utilisabilité du couplage** (*i.e.* de la situation collaborative mise en place). Il nous apparaît important d'ajouter une étape d'évaluation de **faisabilité de l'approche Enactive**. Cette étape doit montrer que l'implémentation réalisée du système met l'utilisateur dans la situation collaborative recherchée. En regard des évaluations de la pertinence, cette évaluation permet de prendre le point de vue de la machine sur l'approche proposée.

Nous présentons dans la suite de cette section les détails des évaluations qui sont à mener pour conclure sur notre travail en les déclinant suivant la première dimension de la méthodologie adaptée :

- **faisabilité** : une machine peut-elle réaliser l'interprétation collaborative selon l'approche proposée ?
- **pertinence** : un utilisateur peut-il utiliser un système fonctionnant sur le mode de collaboration proposé ?
- **utilité/utilisation** : l'approche permet-elle d'envisager une utilisation du système ?

A.1.1 Faisabilité de l'approche Enactive

En premier lieu, l'étape de faisabilité de notre évaluation doit permettre de vérifier que le système développé répond bien aux caractéristiques que nous avons fixées pour réaliser un couplage structurel avec un utilisateur. Dans la mesure, où nous allons nous appuyer sur ce système pour le reste des évaluations de l'approche, cette étape est indispensable pour aborder les étapes suivantes d'évaluation de l'approche Enactive. Ces évaluations sont menées dans un contexte de laboratoire (*i.e.* sans utilisateurs extérieurs au laboratoire, sans objectif applicatif et avec des données spécifiques). Les évaluations sont menées exclusivement sur des critères objectifs.

Nous procédons à l'évaluation de la faisabilité en deux étapes :

1. évaluation des algorithmes de segmentation, classification et apprentissage au regard des motivations de leur choix présentées au chapitre 3.
2. évaluation du système dans sa globalité au regard des pré-requis établis dans le chapitre 2.

Tout d'abord, des évaluations préliminaires sont menées pour les algorithmes de segmentation, de classification et d'apprentissage de scénarios. Nous cherchons par le biais de ces évaluations d'une part à vérifier que les algorithmes accomplissent la tâche qui leur incombe (*i.e.* que les *agents de segmentation* segmentent *bien*, que les *agents de construction de modèles* classent *bien* et que les *agents d'apprentissage*

apprennent *bien*) avec la richesse de l'interprétation qui est attendue. Les choix de ces algorithmes étant motivés en partie par des caractéristiques nécessaires à l'approche Enactive, nous chercherons également à les mettre en évidence.

Dans un second temps, l'évaluation du système dans sa globalité doit nous permettre de conclure sur la vérification de notre système les trois pré-requis :

- le système doit être capable de faire émerger une interprétation des séries temporelles (en incluant la mise en cohérence des annotations et des modèles).
- le système doit montrer une autonomie d'interprétation, en limitant l'introduction de connaissance *a priori*.
- le système doit adapter les modèles au cours de l'interprétation.

En particulier, nous nous intéressons à une expérimentation permettant de montrer la capacité du système à produire la cohérence nécessaire à une interprétation. Nous avons conçu la construction d'une interprétation sur la base de processus très dynamiques et cycliques. Montrer la possibilité de produire de la cohérence consiste alors à montrer la capacité du système à converger. Si le système converge, *i.e.* que tous les agents n'ont plus rien à faire, alors l'interprétation qui a été produite est cohérente (*cf.* section A.2 du chapitre 4). Par définition de nos processus, le système a convergé lorsque :

1. les modèles permettent de retrouver les mêmes annotations qui ont servis à les construire,
2. les annotations permettent de générer les mêmes modèles que ceux qui ont permis de générer ces annotations.

On commencera par évaluer le premier niveau d'interprétation s'articulant autour des modèles d'événements (*cf.* figure 2.4 de chapitre 2). On cherchera à montrer que les modèles d'événements et les annotations automatiques par des segments et des symboles co-évoluent pour converger vers une traduction symbolique *cohérente* des séries temporelles. Ensuite on évaluera l'apport de l'apprentissage et la reconnaissance de scénarios pour la convergence de l'interprétation.

Nous ne proposons pas une preuve formelle de la convergence du processus du système, mais nous évaluons cette propriété souhaitée sur des données simulées. En effet, l'utilisation de données réelles ne peut pas permettre de faire cette étape d'évaluation car les cliniciens ne sont pas en mesure de nous affirmer l'existence de la cohérence recherchée dans les données (*i.e.* existence de modèles d'événements et de scénarios, *cf.* paragraphe B.2.1 du chapitre 1) et de plus, nous souhaitons également nous affranchir dans cette évaluation de la nature des modèles qui ont été utilisés. La simulation de données permet (1) de nous assurer de l'existence d'une cohérence possible dans les données puisqu'elles seront construites sur ce principe et (2) de nous assurer que les modèles utilisés seront capables de capturer les traits caractéristiques des événements dans les données.

À partir des données simulées, on cherchera à retrouver les modèles sous-jacents à leur simulation, et ainsi de montrer que le processus d'interprétation permet de retrouver le *sens caché*¹ dans les séries temporelles. Cependant, il est évident que dans cette évaluation nous ne pourrions juger de l'originalité du point de vue que peut apporter la machine sur les données. Cette partie de l'évaluation est réservée à des applications plus concrètes du systèmes, *i.e.* avec des données réelles.

D'autre part, on sera en mesure de mettre en évidence l'amélioration progressive de l'interprétation, et ainsi de juger de la *bonne* adaptation du système au cours de son expérience.

Parmi les pré-requis du système, nous avons également indiqué que l'intégration d'annotations extérieures participait à la collaboration selon le couplage structurel. Pour autant, nous ne l'évaluons pas dans la faisabilité du système. En effet, les annotations du clinicien s'intègrent de fait aux autres annotations du système. Cette étape correspond plus aux évaluations préliminaires qu'à l'étude du couplage. La pertinence des échanges au moyen d'annotations fait partie des évaluations de la pertinence de l'approche de la collaboration. De plus, nous ne reporterons pas dans ce manuscrit les évaluations techniques préliminaires qui ont été nécessaires à la mise en place des algorithmes (*e.g.* choix des paramètres qui permettent d'obtenir de bons résultats sur nos données).

¹Le processus de simulation de séries temporelles nous permettra de cacher dans les séries temporelles des événements et des scénarios. C'est ce que nous entendons ici par *sens caché*.

A.1.2 Pertinence de l'approche Enactive

L'évaluation de la pertinence de notre approche peut s'évaluer par les performances du couplage entre le système et un humain. On procède pour cela à des évaluations dans un contexte de laboratoire sur des données simulées avec des utilisateurs cliniciens ou non. Les critères d'évaluation sont à la fois objectifs et subjectifs.

L'objectif de ces évaluations est de savoir si le mode de collaboration (incluant les interactions par les annotations) est pertinent pour réaliser l'interprétation de séries temporelles, et en premier lieu si il permet de réaliser une interprétation des données *meilleure* en comparaison d'une interprétation que l'utilisateur réaliserait seul ou avec des modes de collaborations simplifiés.

Nous proposons pour cela une évaluation centrée sur une tâche d'interprétation que nous demandons de réaliser à l'utilisateur sur des jeux de données simulés. Ceci nous permet d'avoir des jeux de données de même *difficulté*², et d'avoir des évaluations quantitatives des performances du couplage. D'une part, l'activité d'interprétation de l'utilisateur est tracée par le système de sorte à quantifier les facteurs de performance (*e.g.* nombre d'annotations réalisées). D'autre part, les résultats de l'interprétation sont comparés avec l'interprétation attendue, *i.e.* celle qui a servi à générer les données.

L'un des objectifs de l'interprétation est de permettre d'accélérer le processus d'annotation de séries temporelles. On aurait pu calculer le temps pour l'atteindre un objectif préfixé, mais d'une part pour les données réelles nous ne pouvons pas fixer un objectif, et ensuite devant la complexité de la tâche d'interprétation, nous ne sommes pas en mesure d'assurer qu'un utilisateur puisse atteindre un objectif préalablement fixé. Par conséquent, nous avons préféré fixer le temps d'interprétation et ne pas évaluer le gain de temps obtenu par la collaboration.

Dans un second temps, les utilisateurs font un retour sur les différentes expériences d'interprétation par interview et questionnaire. L'objectif est alors d'extraire des informations sur la qualité ressentie à la fois de l'aide qui a pu être fournie, en particulier de la pertinence estimée des interactions initiées par la machine (annotations introduites) et des informations retournées. De plus, on interroge également l'utilisateur, qui ne connaît pas l'interprétation attendue, sur la qualité estimée de l'interprétation qu'il a réalisée.

A.1.3 Utilisabilité du couplage

Le prototype développé ne permet pas d'envisager des évaluations de l'utilisabilité du couplage dans le cadre de ce travail. Les conclusions attendues de telles évaluations seraient plus sur le logiciel que sur l'approche théorique elle-même et devrait alors montrer son intérêt en terme d'apport pour les systèmes de monitoring.

Il faudrait alors d'une part disposer de moyens matériels suffisants, *i.e.* d'une grande quantité de données et d'un système implémenté professionnellement pour traiter cette masse de données. D'autre part, il faudrait disposer de moyens humains. En particulier, de telles évaluations doivent être réalisées par des cliniciens prêt à investir du temps à explorer les données et à évaluer les résultats de leur interprétation. Une telle évaluation nécessiterait une longue durée d'évaluation pour être démonstrative et ne peut être envisagée qu'après une validation des concepts de couplage.

Nous nous contentons d'une évaluation à partir des résultats que nous avons extraits de notre propre utilisation du système avec des données réelles.

A.2 Prototype du système pour l'évaluation : CATS

Le système multi-agents a été implémenté pour pouvoir réaliser les expérimentations³. L'implémentation a été réalisée en C++ et les API Qt4⁴ et Qwt⁵ ont été utilisées pour les aspects graphiques.

Les efforts d'implémentation ont permis de développer un prototype que nous avons taché de rendre

²Par la connaissance préliminaire des algorithmes, on peut qualifier *a priori* la *difficulté* d'un jeu de données pour le système, mais il ne s'agit pas d'une difficulté d'interprétation qualifiée dans l'absolue.

³Les codes sources sont accessibles sur demande.

⁴Licence et renseignements : <http://www.trolltech.com>. Utilisation de la version 4.3.0.

⁵Librairie OpenSource de Qt Widget pour des applications scientifiques. Licence et renseignements : <http://qwt.sourceforge.net/>. Utilisation de la version 5.0.2.

le plus opérant et proche possible des paradigmes défendus. Ceci était une étape indispensable pour permettre des évaluations sur des interprétations d'enregistrements de patients et pour le faire manipuler par des utilisateurs extérieurs au projet, incluant des cliniciens. Ce prototype a été nommé CATS (*Collaborative Annotation of Time Series*).

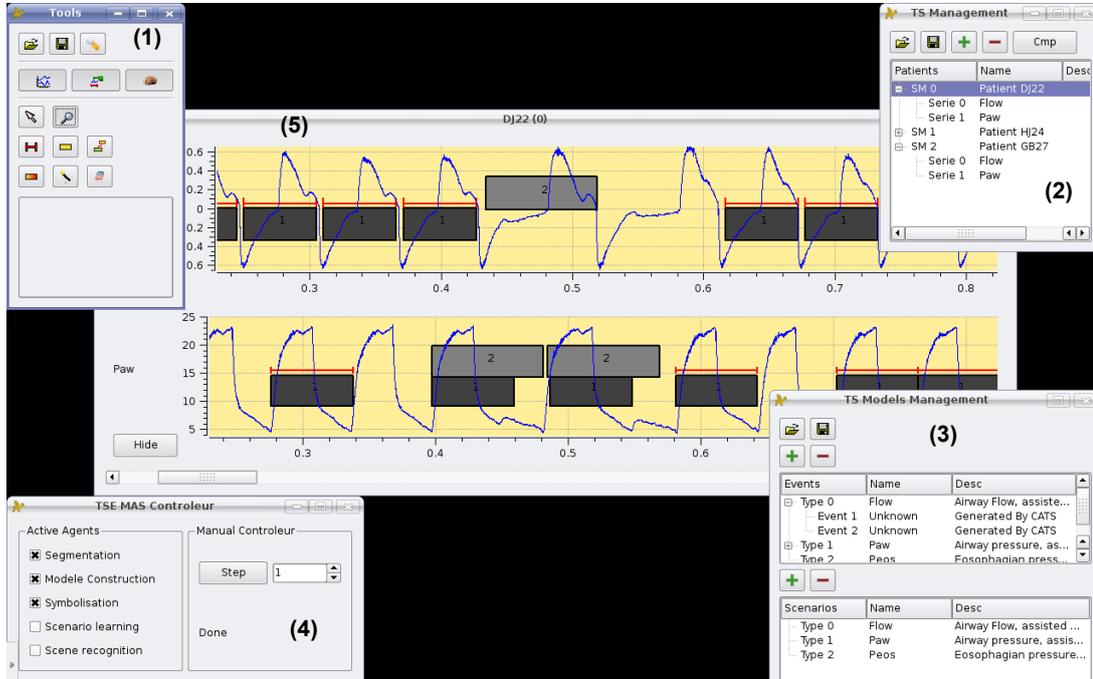


FIG. 4.1.

Interface du système opérationnel pour les évaluations des propositions. L'interface est composée de cinq fenêtres principales : (1) il s'agit de l'interface par laquelle on peut charger les fichiers à annoter et surtout choisir un outil pour modifier une annotation. Les deux fenêtres de droites (2) et (3) sont des fenêtres qui affichent les enregistrements et les séries en cours de traitement pour l'un et les modèles construits et utilisables pour l'annotation. La fenêtre (4) est la fenêtre d'activation des traitements automatiques. Enfin, (5) est une fenêtre dans laquelle on peut parcourir les séries multivariées et les annoter.

La figure 4.1 présente l'ensemble de l'interface opérationnelle pour les tests du système CATS. Par la suite, nous nous contenterons d'illustrer nos évaluations avec le seul l'élément central qui permet de visualiser les séries temporelles et les annotations sur un enregistrement. Nous donnons un exemple de cet affichage sur la figure 4.2. On distingue sur cette figure les différents types d'annotations qui peuvent être réalisés par notre système et par l'utilisateur. Les barres horizontales rouges indiquent les segments. Leur ordonnée n'a aucune signification. Certains segments peuvent être masqués par des symboles. Les rectangles sont des annotations de symboles, l'identifiant du symbole (un nombre entier) associe le symbole à l'évènement qu'il représente. Visuellement, la position en ordonnée et la couleur (ici en niveau du gris) d'un symbole est représentative de l'évènement associé. Sur l'exemple, il y a 5 évènements différents. Finalement, les scènes sont repérées par des surlignages rouges des symboles qui la composent. Les droites rouges (épaisseur double des segments) entre deux symboles indiquent les liens entre un symbole de l'explication et le symbole à expliquer. Ce dernier symbole étant particulier dans la scène, il est repéré par une croix verte pour être distingué des autres symboles de la scène. Sur l'exemple, on a deux scènes toutes les deux expliquent l'évènement 1 par une explication composée des évènements 4 et 5.

A.3 Données

Nous présentons ici les deux jeux de données qui ont été utilisés pendant nos évaluations. Dans un premier temps, nous présentons le jeu de données réelles, *i.e.* des signaux physiologiques de patients ventilé. Dans un second temps, nous présentons la simulation de jeux de données qui a pour but de s'affranchir des hypothèses d'existence de traces de symboles et de scénarios (*cf.* section B.2 du chapitre 2) puisqu'ils ont été construits sur ces concepts. L'utilisation de ces derniers jeux de données sera principalement pour

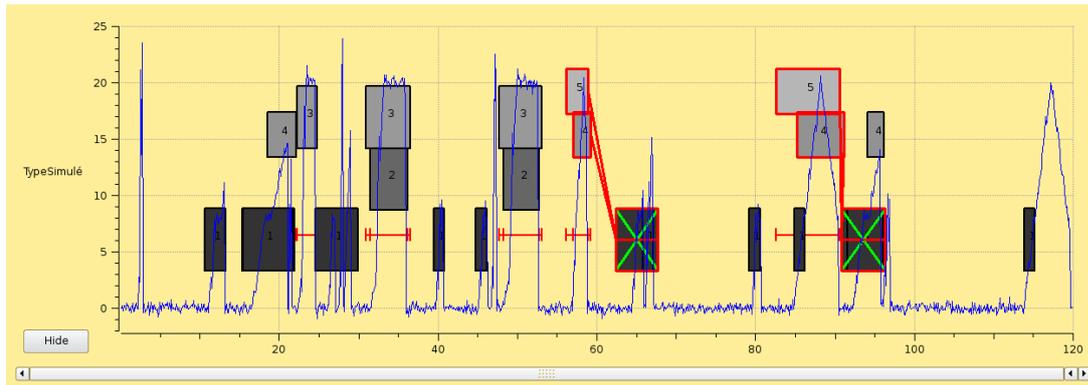


FIG. 4.2.

Illustration des différentes annotations observables sur une série temporelle univariée (ici, une série simulée) : les segments (Barres horizontales), les symboles (rectangles colorés) et les scènes (droites liant des symboles). Le bouton *Hide* sert à masquer l’affichage certaines séries d’une série multivariée et ainsi faciliter la visualisation de deux séries en parallèle. Finalement, la barre de défilement peut être utilisée lorsque seule une partie d’une série temporelle est visualisée (Zoom).

des validations techniques quantitatives.

A.3.1 Données réelles d’asynchronies Patient/Ventilateur de patients en USI

Les patients hospitalisés en unités de soins intensifs et souffrants de désordres respiratoires sont assistés par un respirateur. Le mode dit d’“assistance respiratoire” est un mode de ventilation qui n’automatise pas entièrement la respiration, mais assiste le patient dans son effort inspiratoire. Dès qu’il détecte une inspiration (à l’aide des signaux de débit et de pression) supérieure à un seuil préfixé, il fournit une assistance sous forme de pression ou de volume. De façon similaire, l’aide est stoppée à la détection d’un effort expiratoire. L’objectif d’un tel mode de ventilation est de permettre de conserver le réflexe de respiration spontanée du patient et d’éviter les dysfonctionnements diaphragmatiques induits par des durées de ventilation artificielle prolongées (*i.e.* sur plusieurs jours). Pour cela, une synchronisation entre la machine et le ventilateur est nécessaire pour améliorer le confort du patient et optimiser ses efforts de respiration [182].

Cependant, on constate en pratique des situations d’asynchronies entre le respirateur et le patient. Une asynchronie entre le patient et le ventilateur (notée asynchronie P/V) peut être définie comme une discordance entre les débuts d’inspiration ou d’expiration du patient et de la machine [209]. Thille et *al.* [209] identifient deux types majeurs d’asynchronies P/V (La figure 4.3 illustre ces asynchronies à partir des signaux de pression et débit respiratoire) :

- Les **efforts inefficaces** (*ineffective efforts*) en phase inspiratoire ou en phase expiratoire : ce type d’asynchronies représente 85% des asynchronies rencontrées [209]. Ceci correspond à une situation dans laquelle la machine n’est pas déclenchée à la demande physiologique du patient. Le patient doit alors reconduire sa demande et il fournit ainsi un effort inutile.
- Les **doubles déclenchements** (*double triggering*) : cette situation est rencontrée lorsque le ventilateur se déclenche deux fois successivement, *i.e.* dans un intervalle de temps inférieur à la moitié de la durée moyenne d’un cycle respiratoire du patient.

Les conséquences des asynchronies P/V pendant la ventilation mécanique sont peu connues. L’étude récente de Thille et *al.* montre que sur 60 patients ventilés artificiellement, 24% d’entre eux exhibent un indice d’asynchronies supérieur à 10% d’efforts inefficaces. De plus, ces patients doivent être ventilés artificiellement plus longtemps que les patients ne présentant pas d’asynchronies. Il est cependant nécessaire de prolonger les études pour mettre en évidence plus clairement un éventuel lien de causalité. L’allongement de la durée de la ventilation est peut être lié à la présence d’asynchronies, mais on peut aussi envisager que ce sont les patients dont l’état nécessite une durée de ventilation prolongée qui présentent plus d’asynchronies. Dans les deux cas, il faut d’une part être capable de les détecter pour adapter le mode de ventilation et éviter au patient de les subir et d’autre part essayer de les réduire en tâchant d’améliorer les méthodes de déclenchement des ventilateurs.

Pour cette seconde perspective, des solutions alternatives sont envisagées consistant à s’intéresser à des

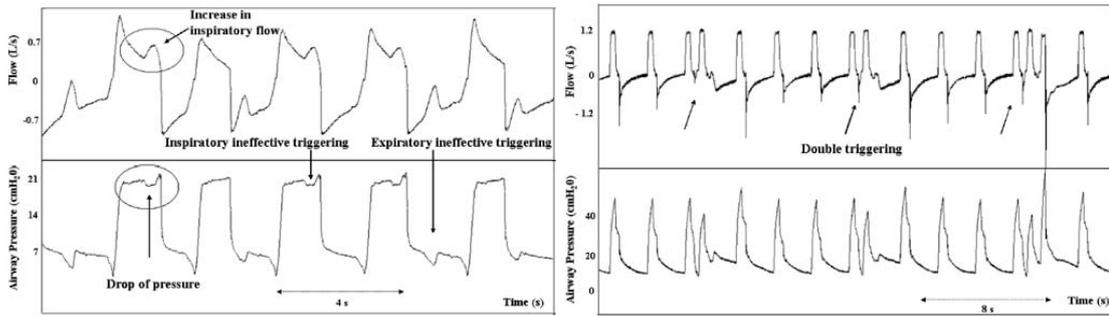


FIG. 4.3. Signaux de pression et de débit respiratoire de patients ventilés en assistance respiratoire et présentant des asynchronies. L'enregistrement de gauche illustre des **efforts inefficaces** en phases inspiratoires et expiratoires. L'enregistrement de droite illustre le cas d'asynchronies de type **double déclenchements**.(Figure extraite de [209]).

signaux en lien plus direct avec la demande du patient. Par exemple, des signaux permettant de détecter les contractions des ailes du nez [5] ou les signaux diaphragmatiques au moyen d'électrodes superficielles sont porteurs d'une information nerveuse des débuts d'inspiration et d'expiration [197].

La détection des asynchronies à partir des signaux nécessite une caractérisation des traces des asynchronies. Cette tâche ardue nécessite une exploration attentive, mais très coûteuse en temps pour des cliniciens experts. Par exemple, la caractérisation graphique des asynchronies à partir des signaux de Nilsestuen et *al.* [159] a dû nécessiter beaucoup de temps d'observation. Notre système doit aider à la réalisation de ces caractérisations sous la forme d'événements et de scénarios.

Par la suite, les caractérisations réalisées pourront servir à l'amélioration de méthodes de déclenchement des ventilateurs. Pour limiter les asynchronies le mode de ventilation doit être continuellement adapté en fonction de l'évolution de l'état du patient, et en particulier pendant son sommeil. En pratique, cette adaptation ne peut pas être réalisée par le personnel de soin car cela nécessiterait une présence permanente et une capacité de détecter les asynchronies. Ceci motive la conception d'un système automatique pour l'adaptation continue de l'assistance lors de la détection d'asynchronies. Il est par conséquent important (1) de détecter automatiquement les asynchronies, (2) d'identifier les facteurs inducteurs d'asynchronies, (3) d'identifier les groupes de patients qui sont plus à même de subir des asynchronies et (4) d'optimiser automatiquement les paramètres de ventilation en fonction des asynchronies détectées et du groupe de patients d'appartenance afin de minimiser les discordances entre patient et ventilateur.

Pour nos expérimentations, nous disposons de 10 enregistrements de patients ventilés. Pour chacun des patients, les signaux de pression et débit ont été enregistrés pendant 30 minutes avec une fréquence d'échantillonnage de 200Hz . Seules les parties les plus stables des enregistrements ont été conservées pour disposer de données de bonne qualité. Chaque série temporelle contient ainsi 64000 points et en moyenne 80 cycles respiratoires. La figure 4.4 donne l'aperçu d'un enregistrement. Plus de détails sur le protocole d'acquisition peuvent être trouvés dans [209].

L'intégralité des signaux a été préalablement manuellement annotée par un clinicien. Pour réaliser cette annotation, le clinicien s'est appuyé sur un signal de pression œsophagienne. Ce signal permet d'identifier de manière objective les asynchronies et ceci indépendamment des signaux de pression et débit. C'est signaux sont rarement utilisés en clinique car il nécessite l'utilisation d'une sonde œsophagienne invasive. Il ne peuvent donc être utilisés pour caractériser les asynchronies par un système de monitoring. L'objectif des cliniciens avec d'un tel jeu de données est d'être capable de caractériser les différentes asynchronies et d'éventuelles traces annonciatrices de leur apparition à partir des signaux de débit et pression.

Dans le cadre de nos évaluations, un tel jeu de données présente l'intérêt d'être à la fois "simple" et "intéressant". Il est "simple" au regard de d'autres signaux physiologiques :

- la périodicité du signal permet de disposer de données pour lesquels des événements peuvent se répéter dans des contextes similaires **et sur les mêmes patients**. En effet, même dans des contextes similaires, les signaux physiologiques de patients différents peuvent être également très différents. Pour nos méthodes d'extraction automatique d'événements et de scénarios, seule une très grande quantité de donnée aurait permis de prendre en compte cette variabilité. Ici, avec une plus faible

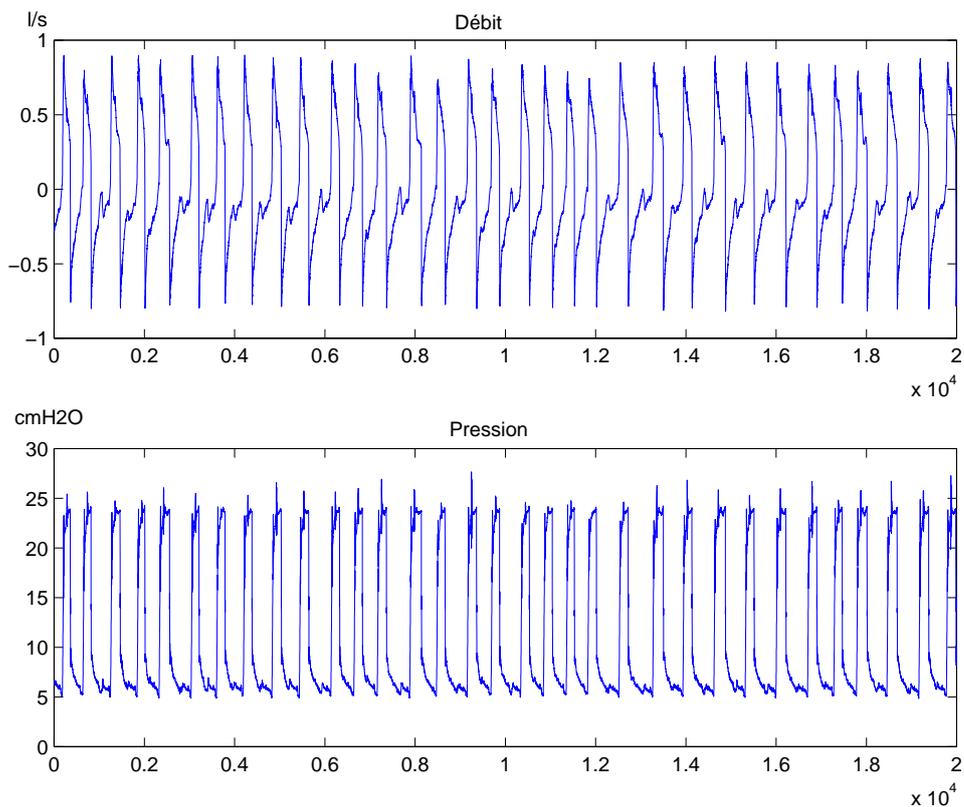


FIG. 4.4. Exemple d’enregistrement d’un patient ventilé artificiellement et présentant des asynchronies. Le signal au dessus est le signal de débit respiratoire et le signal dessous est le signal de pression.

variabilité dans les données, on peut se focaliser sur une plus petite quantité de données.

- la qualité d’enregistrement (haute fréquence et sélection des cycles les plus réguliers) permet de disposer de données peu bruitées.

De plus, ce jeu de données est “intéressant” car il présente un intérêt à être interprété par des cliniciens. L’interprétation permettrait d’une part de caractériser les différents cycles d’asynchronies à partir des signaux de pression et débit respiratoire et d’autre part d’identifier des explications à l’occurrence des asynchronies. Il n’est pas dans les objectifs de ce travail d’apporter des réponses à ces questions complexes, mais nous mettons en évidence qu’il ne s’agit pas d’un “problème jouet”.

A.3.2 Données simulées

En complément des données réelles, nous utilisons des données simulées pour évaluer notre approche. La simulation des données (ici des séries temporelles) est une problématique difficile. Pour plus de détails sur ces difficultés, on pourra par exemple consulter le travail de Duchêne [87]. Notre approche de la simulation est pragmatique, et nous ne cherchons en aucun cas à simuler des données pouvant être rapprochées de données physiologiques tels que les signaux de pression et débit vus précédemment.

Nous cherchons à produire des séries temporelles à partir des notions d’évènements et de scénarios.

Il est critiquable de simuler des données qui correspondent à ce que l’on recherche au moyen de nos propres méthodes. Mais l’approche expérimentale du système introduit déjà beaucoup de difficultés dans une tâche d’interprétation de séries temporelles déjà ardue. Par conséquent, nos évaluations devant essayer de valider notre système consistent plus à montrer que le système ou la collaboration permettent déjà de retrouver des interprétations sous la forme d’évènements et de scénarios dont nous maîtrisons les caractéristiques. On pourra alors envisager l’utilisation d’un tel système sur des données réelles pour lesquelles nous supposons qu’il existe ce type d’interprétation.

De plus, l’utilisation de données simulées va permettre de procéder à des évaluations quantitatives. On

va en effet être capable de quantifier certaines caractéristiques des données, *e.g.* les perturbations dans les données simulées, mais également de quantifier la qualité du résultat obtenu par le processus à partir de mesure du résultat confronté à un résultat objectif. Finalement, l'utilisation de données simulées va permettre d'évaluer le système sur des signaux non-périodiques.

Processus inverse Le processus de simulation est un processus inverse de celui de l'abstraction des séries temporelles de la figure 2.4 : séries temporelles \rightarrow séries symboliques \rightarrow scènes. C'est-à-dire que nous partons d'un scénario que nousinstancions sous la forme d'une scène. Sur la figure 4.5, le scénario est représenté sur le premier graphique en haut⁶. Chaque couleur correspond à un type d'évènements. Plusieurs scènes sont générées comme autant d'exemples du scénario. Les 5 graphiques sous celui du scénario sont des déclinaisons du scénarios sous la forme de 5 scènes (produits grâce à des perturbations du scénario).

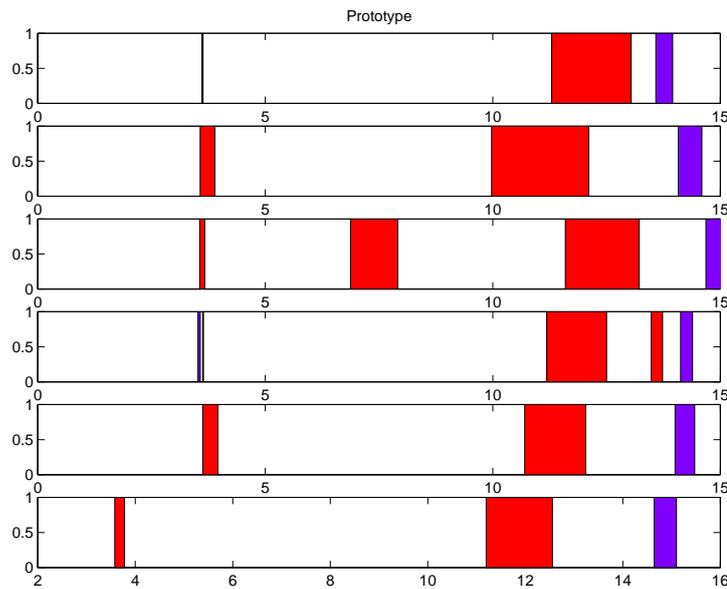


FIG. 4.5.

Simulation de scènes à partir d'un scénario. Sur la première ligne, on visualise le scénario de base et en dessous 5 variantes qui ont été générées en introduisant des perturbations structurelles et temporelles.

Une scène indique alors la position temporelle et la durée des évènements de l'exemple dans l'enregistrement. À partir de modèles d'évènements, des signaux sont générés pour les évènements. Le reste des signaux est rempli par du bruit gaussien. Ce bruit permet l'introduction d'une variabilité supplémentaire entre des segments d'un même symbole. Pour les séries de la figure 4.6, on a utilisé quatre types de modèles définis par l'échantillonnage de formes construites manuellement.

On simule ainsi plusieurs exemples de séries temporelles univariées. Pour obtenir des séries multivariées, on reproduit le même processus pour chaque type de série souhaité.

Introduction de perturbations Nous introduisons différentes perturbations qui sont chacune associée à un taux d'erreur qui peut être paramétré lors de la génération de données :

- Perturbation des scénarios pour la génération des scènes :
 - o Perturbations structurelles,
 - o Perturbations temporelles,
- Perturbations dans la génération des signaux.

Les perturbations structurelles des scénarios consistent à modifier la description symbolique de celui-ci en introduisant des évènements au scénario (*e.g.* la seconde scène sur le troisième graphique de la figure 4.5) ou en en supprimant. On peut également changer le type d'évènement pour l'une des occurrences

⁶Le simulateur de données est implémenté en MatLab. Le code est disponible sur demande.

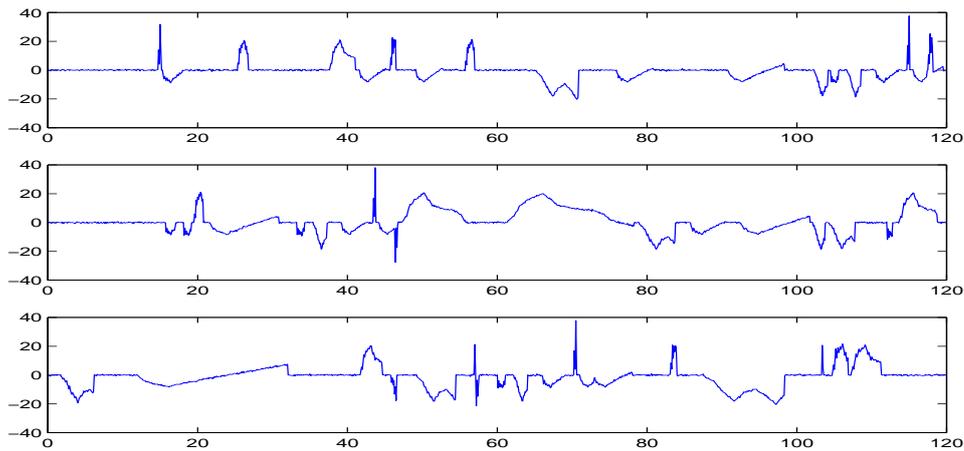


FIG. 4.6. Simulation des séries temporelles à partir des séries symboliques. Les séries symboliques ont été générées à partir de formes de base de chaque évènements auxquelles on rajoute du bruit.

(*e.g.* la troisième scène sur le quatrième graphique de la figure 4.5). Un paramètre de simulation est ajustable : le taux de perturbation (nombre de perturbations à introduire par nombre d'évènements). Pour décider le type de perturbation structurelle, on tire aléatoirement selon une loi uniforme s'il s'agit d'une insertion, suppression ou modification.

Les perturbations temporelles consistent dans un second temps à modifier aléatoirement les valeurs de dates et durées des évènements. Pour cela, les dates et durées des évènements dans un "prototype" de scénario sont associées à des variances. Ainsi la durée et les dates de deux mêmes évènements, tirées selon des lois Gaussiennes, d'un scénario ne sont pas exactement les mêmes entre deux scènes.

Finalement, au niveau de la génération des séries à partir des séries symboliques, on utilise des modèles pour lesquels on peut introduire des modifications aléatoires (*e.g.* leur amplitude).

Paramètres de la simulation Les paramètres de simulation sur lesquels nous pouvons influencer pour produire des exemples ayant des caractéristiques sont de deux types, les paramètres de simulation des séries symboliques à partir d'un scénario et les paramètres de simulation des séries temporelles à partir des séries symboliques.

- Paramètres de la simulation de séries symboliques :
 - o Taille : Nombre d'évènements qui composent un scénario qui sert de prototype à la génération des séries symboliques,
 - o NbreSymb : Nombre d'évènements différents qui peuvent être utilisés.
 - o DateMax/DureeMax : Date et durée maximums des évènements. Permet d'uniformiser la durée globale des séries symboliques. On fixe $DureeMax = DateMax / Taille$.
 - o VarDate : Variance des dates des évènements. Plus la variance est grande, plus les dates seront différentes entre les séries symboliques générées à partir du même scénario.
 - o VarDuree : Variance des durées des évènements. Plus la variance est grande, plus les durées seront différentes entre les séries symboliques générées à partir du même scénario. (On fixe $VarDuree = VarDateMax / Taille$).
 - o ErrStruct : Taux d'erreurs structurelles introduites (ajout, suppression ou modification à égale probabilité).
- Paramètres de la simulation de séries temporelles :
 - o g : Intensité d'un bruit Gaussien ajouté uniformément à la série temporelle.
 - o Les paramètres des modèles spécifiquement utilisés, pour nous il y a uniquement l'amplitude.

Dans la suite de ce chapitre, nous présentons les évaluations menées et leurs résultats en suivant la même structure que lors de la présentation de la méthodologie dans la section A.1. Nous commençons

par donner les résultats de l'évaluation de la faisabilité de l'approche puis ceux de l'évaluation de sa pertinence et enfin nous présentons une utilisation du système pour l'interprétation de séries temporelles pour conclure sur l'utilisabilité du couplage.

B Évaluations de la faisabilité de la collaboration

La faisabilité de l'approche collaborative consiste à expérimenter le système multi-agents pour monter qu'il respecte les pré-requis d'un système permettant un couplage structurel tel que nous l'avons envisagé. Avec des données simulées, nous nous sommes intéressés à l'émergence d'une interprétation pour savoir si (1) les méthodes algorithmiques permettent de réaliser l'interprétation des séries temporelles selon les principes Enactifs et si (2) les mécanismes mis en place permettent de faire converger le système.

B.1 Évaluation des algorithmes

Nous avons proposé des algorithmes de segmentation, classification et d'apprentissage de scénarios implémentés dans les agents spécifiquement pour noter approche de la construction d'une interprétation des séries temporelles. L'objectif de ces évaluations est de valider le choix de ces propositions au regard :

1. de leurs capacités à réaliser la tâche qui leur est assignée.
2. des spécificités attendues pour faire que le système dans sa globalité respecte les principes de l'Enaction que nous avons fixés.

B.1.1 Algorithme de *pattern matching* pour la segmentation de séries temporelles

Expérimentation L'évaluation du *pattern matching* est une évaluation couplée des modèles que nous utilisons pour les événements et de la méthode algorithmique permettant de les retrouver. Pour ces tests, nous utilisons les données d'asynchronies P/V. Nous testons la possibilité de retrouver des différents types de cycles (normaux, efforts inefficaces ou doubles déclenchements) intra-patient et inter-patients. Les évaluations intra-patient permettent d'évaluer la méthode de *pattern matching* sur sa capacité à retrouver des cycles à partir des modèles construits sur quelques annotations. Pour les évaluations inter-patients, les modèles sont construits à partir des annotations sur un patient et évalués sur un autre. On espère ainsi mettre en évidence que les modèles qui sont construits sont suffisamment génériques pour être appliqués sur différents patients. Pour ces évaluations, des travaux pointant la possibilité de caractériser les asynchronies sur les formes des cycles [159], on utilise des modèles qui capturent la forme des segments.

L'évaluation se fait en deux étapes. Dans un premier temps, un modèle construit par des annotations préalables d'un clinicien. Sur la première moitié du signal, on tire au hasard parmi les annotations qui servent à l'apprentissage. Un pourcentage variable d'annotations est utilisé. Dans un second temps, le modèle est utilisé pour l'algorithme de *pattern matching* sur la seconde moitié de l'enregistrement pour les évaluations intra-patient et sur un autre enregistrement pour les évaluations inter-patients. Cette opération est répétée 20 fois pour chaque couple de patients (patient *modèle* et patient annoté, pouvant être les mêmes).

Dans le cas des cycles respiratoires, deux patients ont des durées de cycle très différents. Par conséquent, on peut modifier manuellement les durées minimums et maximums pour quelles correspondent aux durées des cycles respiratoires. Le reste du modèle est inchangé.

On compare alors le résultat de l'annotation réalisée automatiquement avec le résultat de l'annotation préalable du clinicien. On peut alors identifier les annotations vraies positives (VP), les faux positifs (FP), les faux négatifs (FN) et les vrais négatifs (VN)⁷. Ces valeurs permettent de calculer alors la spécificité et le sensibilité⁸ de la méthode (modèle + algorithme).

Résultats Globalement, les résultats des tableaux 4.1 et 4.2 montrent que l'algorithme de *pattern matching* offre une spécificité acceptable, et une sensibilité importante. Ces résultats sont en accord avec l'utilisation qui est faite de la méthode pour la segmentation des séries temporelles. En effet, la forte

⁷Dans la mesure où on dispose d'un nombre fixé de cycles, les vrais négatifs peuvent être comptabilisés. Si on annoté des asynchronies, VN est donné par le nombre de cycles qui n'ont pas été considérés à raison comme des asynchronies.

⁸Les définitions de la spécificité et de la sensibilité peuvent être trouvées en section A.1.2 du chapitre 1.

<i>Données</i> <i>Modèle</i>	DJ22	HJ24	GB27
DJ22	0.57	0.71	0.54
HJ24	0.82	0.80	0.50
GB27	0.68	0.72	0.71

TAB. 4.1.

Résultats croisés pour les évaluations du *pattern matching* sur trois patients : La table donne les valeurs de *Specificite*. Sur la diagonale, on retrouve la moyenne des résultats des expérimentations intra-patient et sur le reste de cette matrice, les résultats inter-patients.

<i>Données</i> <i>Modèle</i>	DJ22	HJ24	GB27
DJ22	0.87	0.98	0.85
HJ24	0.88	0.95	0.89
GB27	0.98	0.98	0.97

TAB. 4.2.

Résultats croisés pour les évaluations du *pattern matching* sur trois patients : Sensitivité. Sur la diagonale, on retrouve la moyenne des résultats des expérimentations intra-patient et sur le reste de cette matrice, les résultats inter-patients.

sensitivité nous montre que la méthode va permettre de retrouver une grande partie des motifs similaires à un modèle et ce malgré la variabilité des durées des motifs (prise en compte de la dilatation des motifs). Par conséquent, l'utilisateur n'aura que peu d'ajouts à réaliser pour compléter une annotation automatique de la machine. En ce qui concerne la spécificité, entre 30% et 50% des segments se voient attribuer un mauvais évènement (FN). Cette modeste spécificité nous indique que la méthode risque de produire des segments non-attendus.

Ces résultats relativement bons sont à mettre au crédit de la possibilité de retrouver des motifs de tailles différentes. En particulier pour les évaluations inter-patients, si les algorithmes et les modèles ne sont pas capables de prendre en compte des durées variables des formes de segments à retrouver, les résultats chutent.

On constate logiquement qu'en moyenne les résultats intra-patient sont meilleurs que les résultats inter-patients. Cependant, on peut constater que le modèle de GB27 dépasse tous les autres modèles en terme de sensibilité. On peut expliquer ce résultat par le fait que l'enregistrement GB27 comportait beaucoup d'asynchronies de types efforts inutiles (89 asynchronies contre seulement 11 cycles normaux également réparties sur la durée de l'enregistrement) par conséquent, le modèle du motif d'asynchronie était plus générique que tous les autres donnant une plus grand sensibilité. En contrepartie, les résultats obtenus par les autres modèles sont moins bons lorsqu'ils utilisent cet enregistrement comme donnée d'évaluation car le grand nombre d'asynchronies leur donnait d'autant plus de chance de se tromper. D'où une forte spécificité des modèles DJ22 et HJ24 appliqués aux données de GB27.

HJ24 est l'enregistrement qui permet d'obtenir en moyenne les meilleurs résultats sur les trois patients. C'est l'enregistrement qui présente le moins de bruit et des motifs visuellement très différenciés entre les cycles d'asynchronies et d'asynchronies.

La courbe ROC de la figure 4.7 illustre une expérimentation permettant de mettre en évidence l'importance du nombre d'annotations dans la construction d'un modèle de cycle. L'expérimentation a été réalisée comme précédemment (intra-patient) en faisant varier le nombre d'annotations qui ont servies à construire un modèle de cycle asynchronie. Les spécificités et sensibilités ont été calculées pour des constructions de modèles à partir de 30%, 15%, 10%, 5% et 2% de l'ensemble des cycles d'un enregistrement. Les points de la figure 4.7 indiquent les valeurs moyennes et les hauteurs et largeurs des boîtes indiquent les variances des deux mesures. L'aire sous la courbe calculée est de 0.88. Ceci illustre les bonnes performances globales de la méthode. L'allure de la courbe montre également logiquement que plus le nombre d'annotations réalisées est grand, plus la méthode est sensible et moins elle est spécifique.

Conclusions En conclusion, les résultats de la performance brute de l'algorithme de *pattern matching* a une sensibilité importante et une spécificité acceptable. L'obtention de meilleures performances passe

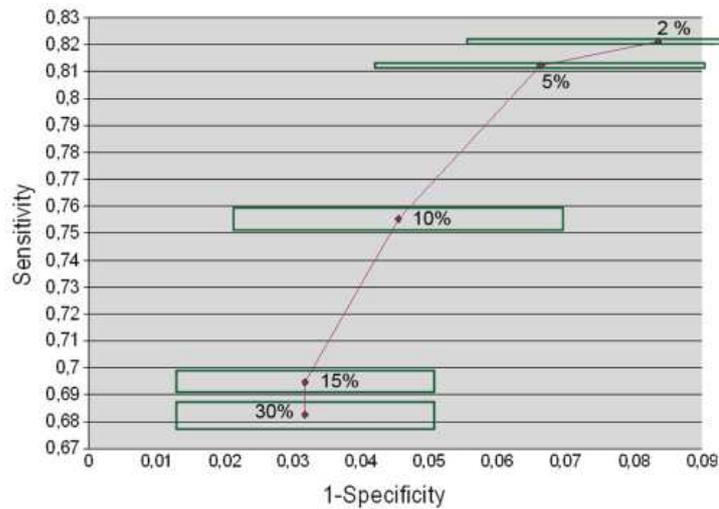


FIG. 4.7.

Évaluation du *pattern matching* : Courbe ROC. On reporte sur ce graphique les résultats de spécificité et sensibilité obtenus pour des constructions de modèles à partir de 30%, 15%, 10%, 5% et 2% de l'ensemble des cycles d'un enregistrement. Les points indiquent les valeurs moyennes et les hauteurs et largeurs des boîtes indiquent les variances des deux mesures.

par le choix d'un formalisme de modèles plus adapté à des données particulières.

Pour les *agents de segmentation* qui servent d'"organe de perception" au système en l'alimentant avec les éléments de base de l'interprétation de séries temporelles (les segments). La segmentation doit permettre :

1. d'être capable d'identifier des segments à partir de modèles expressifs,
2. de s'adapter au cours de l'interprétation,
3. d'assurer une cohérence entre la segmentation et les évènements.

Pour le premier point, l'un des intérêts de l'algorithme est de reporter une partie de cette question sur le modèle de segment (et la distance qui y est associée) qui est utilisé. L'algorithme permet de prendre en compte d'explorer des segments de tailles différentes, mais la sélection d'un segment est principalement dépendante du modèle et du formalisme utilisé.

En l'occurrence, nous avons utilisé pour cet évaluation un modèle de forme, avec longueurs variables. Nos évaluations sur cet aspect sont finalement assez limitées puisqu'elles permettent simplement de mettre en évidence que 6 formes différentes (les 6 formes typiques existantes : normal, double déclenchement et effort inefficace sur les signaux de pression et de débit). Avec notre formalisme de modèles, nous n'avons pas été limité pour représenter ces différents évènements, mais il est certain qu'un tel formalisme peut rapidement être insuffisant. Il se trouve que pour ces données, nous avons la connaissance *a priori* qu'un tel modèle était adéquat et c'est pourquoi nous l'avons préféré à l'implémentation d'une représentation avec des caractéristiques de segments⁹. Et de fait, l'utilisation d'un modèle de forme permet d'avoir de bons résultats.

L'expressivité de notre méthode s'illustre également dans la faible spécificité de nos modèles : cette valeur met en évidence que malgré la variabilité naturelle des longueurs de cycle (intra- et inter-patients), l'algorithme permet de retrouver des cycles de tailles différentes. Cette capacité de retrouver des évènements dans les séries temporelles qui n'ont pas nécessairement une durée fixe est un point positif de notre algorithme.

On voit également un point positif à l'introduction erronées de segments par les agents. Ces nouveaux segments peuvent permettre au système de s'intéresser à d'autres types de motifs. Mais il faut pour cela que ces segments puissent être distingués dans la symbolisation et la classification, on peut espérer que le raffinement des modèles permettra de progressivement révéler les segments utiles et de réduire le nombre de segments qui ne le sont pas (*cf.* évaluations de la section B.2).

Pour le second point, la courbe ROC permet de mettre en évidence que le système peut s'adapter au

⁹Des expérimentations supplémentaires permettraient alors de conclure plus largement sur l'expressivité de la méthode avec des modèles de caractéristiques de segments.

cours de l'interprétation puisque les modèles sont modifiés par le nombre d'exemple qui sont utilisés pour construire le modèle. De plus, la courbe montre l'intérêt pour le système d'acquérir une expérience riche puisqu'elle met en évidence que cela permet d'améliorer la caractérisation de ses modèles, *i.e.* de spécifier sa perception des données.

Finalement, la cohérence entre la segmentation et les évènements est directement liée à la sensibilité et la la spécificité de l'algorithme de segmentation, puisque la segmentation sera en cohérence avec les évènements. Les segments qui sont retrouvés par la segmentation sont bien ceux qui sont par le biais du modèle des évènements.

B.1.2 Algorithme de classification

Pour l'évaluation de notre algorithme de classification, on utilise de nouveau les données réelles d'asynchronies P/V. Les annotations du clinicien qui sont à notre disposition servent d'une part à définir manuellement les segments qui serviront d'exemples à classer et d'autre part ils donnent la classe de référence des exemples parmi les trois possibles : "effort inefficace", "double déclenchement" ou "normal". Si notre algorithme (dénommé ici MN) classe correctement les segments, alors il doit être en mesure de retrouver ces trois classes et d'attribuer les segments à la bonne classe.

Nous avons proposé un algorithme qui a les propriétés spécifiques fondées sur l'Enaction :

- construction de distances propres et caractéristique à chaque classe,
- construction couplée de la classification et des classes,
- utilisation de seuls critères de cohérence de la classification.

Plutôt que de vérifier ces propriétés, il nous apparaît plus intéressant de montrer qu'avec celles-ci¹⁰, notre algorithme permet de réaliser une classification sur laquelle notre système peut s'appuyer pour construire des évènements (*i.e.* des classes de segments) à partir des segments. Dans un contexte d'une évaluation de faisabilité du système, il s'agit ici d'évaluer la viabilité de la proposition. **L'algorithme sera viable, s'il montre des performances de classification (rappels et précisions) comparables à d'autres algorithmes de classification.** C'est pourquoi, dans un premier temps, nous comparons les performances de notre algorithme avec celles d'une classification de type K-Means (KM)¹¹.

Ensuite nous avons voulu étudier spécifiquement le comportement de notre classification avec l'ajout de liens par un expert afin de savoir si l'ajout de liens permet bien d'améliorer les résultats. Ce second point constitue une spécificité de cette méthode de classification qui permet d'envisager l'amélioration de classes de segments par un utilisateur couplé au système en conservant l'autonomie de constitution des classes au système requis par l'approche Enactive.

Évaluation des performances de l'algorithme MN face à l'algorithme KM Nous comparons ici les performances de l'algorithme MN face à l'algorithme KM. Le choix de l'algorithme KM parmi les méthodes de clustering non supervisé pour comparer notre algorithme a été fait pour sa proximité algorithmique de notre algorithme MN. On va pouvoir ainsi juger des modifications qui sont apportées.

Pour cette évaluation, nous disposons des données de 9 patients comportant en moyenne 80 cycles, et donc 80 exemples à classer. Les segments de chaque patient sont d'abord traités séparément. Ceci permet de réaliser 9 classifications différentes. Pour la plupart des jeux, il n'y a pas de cycle "doubles déclenchements", seules deux classes sont donc à trouver. Il y a une proportion de 34% d'asynchronies contre 66% d'exemples normaux dans l'ensemble des patients. Entre chaque patient il peut y avoir de grandes disparités d'attribution ce qui fait qu'une classe est généralement surreprésentée.

On calcule ensuite pour chaque classification obtenue le rappel et la précision de cette classification. Pour les K-Means comme pour l'algorithme MN, les clusters construits se voient attribuer un nom de classe ($i \in$ "effort inefficace", "double déclenchement" ou "normal") dont les exemples sont majoritaires dans la classe. Ceci permet de construire des matrices de confusion. Le rappel et la précision d'une classe sont alors donnés par les formules suivantes :

$$Rappel = \frac{\text{Nombre d'exemples de type } i \text{ affecté à la classe du type } i}{\text{Nombre total d'exemples du type } i}$$

¹⁰Pour vérification de ces propriétés, les codes peuvent être fournis.

¹¹Nous avons réimplémenté l'algorithme des K-Means de la manière la plus simple.

$$Precision = \frac{\text{Nombre d'exemples de type } i \text{ affecté à la classe du type } i}{\text{Nombre total d'exemples de la classe}}$$

Finalement, le rappel et la précision de la classification sont obtenus en moyennant les rappels et précisions des classes.

Le modèle de segments utilisé pour la classification est le modèle de forme de taille 10, 20 ou 30. On réalise les expérimentations pour les trois tailles.

Dans la mesure où notre algorithme doit trouver le nombre de classes alors que pour les K-Means disposent d'un nombre de classes *a priori*, nous *pénalisons* l'algorithme de K-Means en réalisant pour chaque classification, des classifications avec un nombre de classe de 1, 2, 3 puis 4. Et on prend la moyenne des rappels et précisions obtenus¹². Le caractère aléatoire de l'algorithme ne donnant pas toujours les mêmes résultats et par mesure de symétrie de l'évaluation, nous réalisons également les moyennes des rappels et précisions sur quatre classifications avec la classification MN.

Finalement, on précise que le paramétrage de l'algorithme de classification MN est resté le même pour toute l'expérimentation et a été empiriquement choisi à partir des données du patient GB27 avec un modèle de 20 points¹³.

La table 4.3 reporte les résultats de cette expérimentation pour les signaux de pression. Pour chaque ligne, il s'agit d'une moyenne obtenue à l'aide de 12 expérimentations (3 tailles de modèles et 4 essais). La taille du modèle n'a pas montré d'influence sur les résultats, c'est pourquoi nous les agglomérons.

	<i>Signal de Pression</i>			
	K-Means		MN Classif	
	Rappel	Précision	Rappel	Précision
BR52	0,75	0,85	0,41	0,92
CDJF28	0,77	0,85	0,38	0,96
DJ22	0,68	0,75	0,32	0,91
GB27	0,52	0,87	0,26	0,85
GJ21	0,75	0,85	0,24	0,96
HJ24	0,59	0,80	0,26	0,93
LM42	0,81	0,92	0,71	0,95
MA29	0,78	0,92	0,34	0,89
NA27	0,52	1,00	0,23	1,00
TOTAL	0.69	0.87	0.35	0.93

TAB. 4.3.

Résultat des expérimentations de classification sur des segments de signaux de pression. Comparaison des Rappels et Précisions de MN et KM.

Les résultats de l'algorithme MN montrent donc un rappel systématiquement moins important que le rappel de l'algorithme K-Means. Dans 8 cas sur 9, le rappel est même au moins deux fois inférieur. Pour le 9ème cas (LM42), l'enregistrement comporte une très large proportion de cycles asynchrones (seulement 6 cycles normaux et 3 cycles de doubles déclenchements), de plus les cycles d'asynchronies sont très réguliers. Pour ces deux raisons, c'est un jeu plus facile à classer que les autres. D'ailleurs, les résultats du rappel de l'algorithme K-Means sont également meilleurs pour ce jeu.

Pour notre algorithme, la présence d'exemples de la classe *Double Déclenchements* semble aider la classification globale en offrant une possibilité d'avoir une classe stable. En effet, les exemples de double déclenchement sont très différents des autres. Ils sont par conséquent plus faciles à regrouper.

Le rappel de notre méthode est globalement moins bon que celui de l'algorithme K-Means car l'algorithme MN a tendance à créer un trop grand nombre de classes. Par exemple, les matrices de confusions des tables 4.4 montrent le résultat des classifications pour HJ24 obtenu avec MN (Rappel 0.12, Précision 0.93) et KM (Rappel 0.55, Précision 0.82). Il s'agit ici d'un exemple pour lequel un nombre très important de classes a été généré par MN, en général, moins de classes ont été générées.

¹²Certaines méthodes permettent de proposer à partir du jeu de données le nombre de classes, mais nous ne les avons pas utilisées. Les résultats de KM seraient améliorés en les utilisant.

¹³Le paramétrage complet de la méthode de classification des cycles de Pression est la suivante : $eps = 5$, $epsComp = 1$, $fusionThreshold = 2$, classification non stabilisée. On pourra se reporter à l'annexe B pour plus de détails sur les paramètres.

En théorie, les classes C_0, C_1, C_2, C_3 et C_4 obtenues avec MN ne devraient pas être séparées puisqu’elles ne contiennent que des exemples de la première classe. Il en est de même pour les classes C_5 et C_6 . Le rappel de chacune de ces classes étant faible, la moyenne des rappels qui nous sert de mesure est également faible. En revanche, pour KM avec un choix de deux classes, le rappel est plus important. De plus, l’algorithme MN peut engendrer le non classement d’exemples. En cas de bouclage empêchant la stabilisation d’une classification, des exemples sont mis de côtés réduisant ainsi le rappel.

	<i>Effort inefficace</i>	<i>Normal</i>		<i>Effort inefficace</i>	<i>Normal</i>
C_0	31	1			
C_1	4	0			
C_2	2	0			
C_3	1	0	C_0	25	1
C_4	1	0	C_1	10	12
C_5	0	4			
C_6	0	2			
C_7	1	1			

TAB. 4.4.

Comparaison des matrices de confusion obtenues pour les classifications MN (gauche) et KM (droite) du jeu de données HJ24. Chaque ligne donne les nombres de cycle “effort inefficace” et de cycle “normaux” que contient une classe C_i trouvée par l’algorithme. Le grand nombre de classes obtenues par MN explique le faible rappel.

Dans le cadre de notre mesure, la précision est en moyenne meilleure que pour les K-Means, bien que sur certains jeux de données (GB27, MA29) les résultats de KM sont meilleurs que ceux de MN. Par contre, si on prend le maximum des précisions obtenues sur les 4 essais pour les K-Means, ce qui revient à lui faire faire le choix du meilleur nombre de classes sur le critère de précision, alors les résultats sont du même ordre que ceux des MN. Le jeu de données NA27 donne des précisions de 1 pour les deux classifications car les seules annotations utilisées sont des cycles d’asynchronies (il n’y a donc qu’une classe dans ce jeu).

La création d’un grand nombre de classes explique de nouveau la bonne précision de notre méthode. En effet, chaque classe obtenue est plus “pure” que si elle avait été mélangée en risquant d’être trop générale et ainsi inclure des exemples non-conformes. Les matrices de confusion du tableau 4.4 montrent de nouveau ce résultat.

En résumé, malgré les mesures et le protocole qui peuvent pénaliser l’algorithme de K-Means, ces résultats montrent que l’algorithme MN restent moins performant que l’algorithme KM.

On peut indiquer que les temps de calculs entre les deux algorithmes sont équivalents et négligeables avec des classifications de moins de 100 exemples. En particulier, si la paramétrisation de l’algorithme MN n’est pas bien réalisée, l’algorithme peut avoir des difficultés à stabiliser sa classification. Ce genre de comportement n’a pas été rencontré à partir d’une paramétrisation réalisée sur un jeu de données et réutiliser sur les autres. Ceci permet de penser que la paramétrisation d’un type de série peut être utilisée pour plusieurs patients.

De plus, bien que la paramétrisation de l’algorithme MN ait été faite sur le patient GB27, ces résultats sont inférieurs aux autres. Nous expliquons cela par le fait que GB27 est un enregistrement sur lequel il y a beaucoup d’artefacts qui peuvent engendrer des perturbations dans les modèles.

En conclusion, les résultats obtenus montrent que cette classification conçues sur les principes de l’Enaction et qui permet au système d’adapter les classes de segments (et donc les événements) au cours de l’interprétation sur un principe de cohérence peut être envisagée comme une solution viable pour les *agents de classification*.

On a pu voir que l’origine des moindres performances est le nombre de classes engendrées par l’algorithme. Nous espérons qu’en ajoutant des liens entre les exemples, deux classes comportant des exemples *a priori* identiques pourront fusionner plus facilement et ainsi améliorer le rappel de l’algorithme MN.

Évolution des performances avec l'ajout d'annotations À la suite de ce que nous venons d'indiquer nous cherchons à savoir si l'ajout d'annotations, *i.e.* des liens entre deux exemples, va permettre d'améliorer le rappel des classifications de MN.

Pour cela, nous mesurons l'évolution du rappel et de la précision en fonction du nombre de liens introduits. Pour chacun des 9 jeux de données précédents (Segments annotés de signaux de pressions de 9 patients), nous réalisons des classifications à l'aide de l'algorithme MN en ayant ajouté préalablement de 1 à 30 liens différents entre des exemples de classes *a priori* similaires (typiquement deux exemples d'asynchronies ou bien deux exemples de non-asynchronies liés ensemble). Ces liens sont tirés au hasard. Nous calculons ensuite le rappel et la précision des classifications obtenues. Pour chaque expérimentation (jeu de données et nombre de liens) nous donnons dans les tableaux 4.5 et 4.6 les moyennes des mesures sur les 9 classifications et nous l'illustrons par la figure 4.8.

Nb liens	DJ22	HJ24	GB27	MA29	GJ21	BR52	LM42	CDFJ28	NA27	Moy.
De 1 à 5	0,41	0,34	0,46	0,49	0,36	0,67	0,84	0,54	0,26	0.49
6 à 10	0,42	0,34	0,48	0,57	0,42	0,6	0,82	0,45	0,38	0.50
11 à 15	0,43	0,39	0,54	0,71	0,42	0,6	0,67	0,57	0,61	0.547
16 à 20	0,46	0,4	0,46	0,68	0,43	0,61	0,77	0,52	0,66	0.555
21 à 25	0,54	0,45	0,47	0,69	0,42	0,6	0,77	0,56		0.561
26 à 30	0,51	0,42	0,42	0,66	0,48	0,61	0,72	0,62		0.555

TAB. 4.5. Données d'évolution du rappel en fonction du nombre de liens pour chaque patient et en moyenne.

Nb liens	DJ22	HJ24	GB27	MA29	GJ21	BR52	LM42	CDFJ28	NA27	Moy.
De 1 à 5	0,87	0,9	0,81	0,85	0,96	0,94	0,95	0,94	1	0.91
6 à 10	0,86	0,9	0,87	0,88	0,95	0,95	0,92	0,94	1	0.92
11 à 15	0,83	0,9	0,84	0,9	0,95	0,94	0,88	0,96	1	0.91
16 à 20	0,8	0,89	0,9	0,93	0,94	0,95	0,89	0,94	1	0.92
21 à 25	0,8	0,88	0,86	0,94	0,96	0,97	0,92	0,95		0.91
26 à 30	0,8	0,9	0,9	0,91	0,95	0,95	0,88	0,96		0.91

TAB. 4.6. Données d'évolution de la précision en fonction du nombre de liens pour chaque patient et en moyenne.

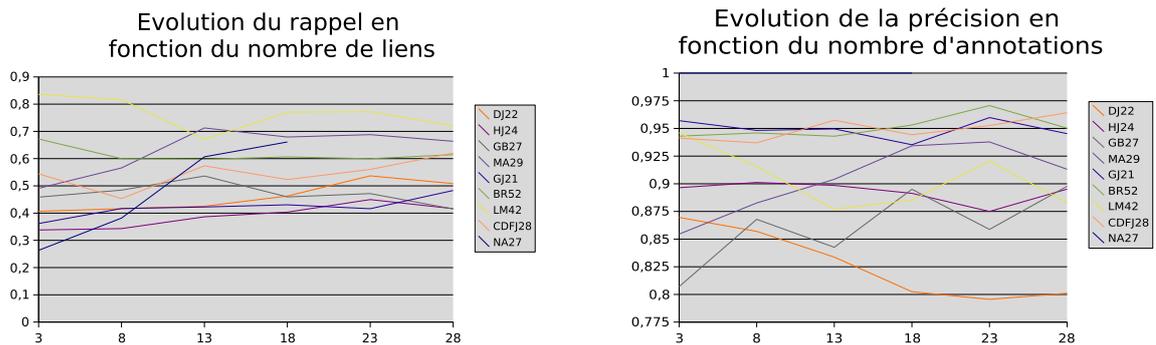


FIG. 4.8. Évolution du rappel (gauche) et de la précision (droite) en fonction du nombre de liens dans notre annotation.

La figure 4.8 résume les tableaux 4.5 et 4.6. On peut tout d'abord voir que le nombre de liens n'influence pas la précision de l'algorithme MN. Pour le rappel, on peut noter numériquement une légère amélioration en moyenne en fonction du nombre de liens : On passe de 0,48 avec 1 à 5 annotations à 0,55 avec plus de 25 annotations. Prise indépendamment, on peut constater que deux types de comportement. Les rappels de BR52, CDFJ28 et LM42 régressent initialement puis se stabilisent. Par contre, pour tous les autres, on constate une évolution "importante" entre 1 et 15 annotations puis une stabilisation ou un ralentissement

de l'amélioration. Si on calcule les moyennes des rappels sans BR52, CDFJ28 ni LM42 on passe de 0,39 à 0,52, soit plus de 25% d'amélioration du rappel par l'ajout de liens entre les exemples.

On peut donc conclure qu'en moyenne, le rappel est amélioré par l'introduction de liens et que la précision reste à un même niveau élevé. Cependant, il faut bien noter que les résultats de cette expérimentation ne permettent pas de mettre en évidence de manière flagrante cette amélioration, même si les liens sont ajoutés en grand nombre.

Suite à la comparaison avec les K-Means, nous avons indiqué que le nombre de classes était la raison du faible rappel de notre méthode. Nous avons alors réalisé de nouvelles expérimentations sur 6 jeux de données pour vérifier si l'amélioration du rappel était liée à la réduction du nombre de classes. Nous donnons les résultats dans la table 4.7. On peut ainsi constater qu'en moyenne on a une réduction du nombre de classes et que sur chaque exemple, la réduction du nombre de classes est corrélée avec la réduction du rappel¹⁴ : GJ21, MA29, DJ22 et HJ24 réduisent notablement leur nombre de classes, mais BR52 et CDFJ28 les réduisent très peu.

<i>Nb liens</i>	<i>BR52</i>	<i>GJ21</i>	<i>MA29</i>	<i>CDFJ28</i>	<i>DJ22</i>	<i>HJ24</i>	Moy.
<i>De 1 à 5</i>	5	6,14	6,1	4,66	4,78	5,32	5,33
<i>6 à 10</i>	4,68	5,4	6,18	4,68	4,64	5,24	5,14
<i>11 à 15</i>	4,68	4,78	5,28	4,28	3,86	4,82	4,62
<i>16 à 20</i>	4,82	4,54	4,78	4,34	4,74	5,4	4,77
<i>21 à 25</i>	4,34	4,78	4,94	4,76	4,76	5,46	4,84
<i>26 à 30</i>	4,66	4,58	4,18	4,32	3,6	4,62	4,33

ТАВ. 4.7.

Données d'évolution du nombre de classes en fonction du nombre d'annotations pour chaque patient et en moyenne.

En conclusion, on peut indiquer que l'algorithme peut être utilisé pour prendre en compte les interactions d'un utilisateur qui ajouterait des liens entre deux exemples, en effet :

- les expérimentations montrent que ces liens permettent à un utilisateur d'améliorer la classification, puisqu'il améliore le principal défaut de notre algorithme qui est de constituer trop de classes (qui sont trop spécifiques)
- l'introduction de connaissance *a priori* sous la forme de liens n'est pas un frein pour la constitution des classes par l'algorithme de classification, et permet donc au système de conserver toute son autonomie de construction d'évènements.

B.1.3 Algorithme d'Apprentissage et Reconnaissance de scénarios

Nous avons indiqué que le choix des algorithmes d'apprentissage et reconnaissance de scénarios étaient principalement motivé par leur apport pour l'expressivité de l'interprétation. Les évaluations de ces algorithmes doivent montrer que les scénarios construits sont riches et en particulier que la dimension temporelle peut être prise en compte. Nous réalisons donc pour cela des évaluations qui permettent de mettre en évidence les capacités de notre algorithme à construire de scénarios expressifs.

De plus, dans la construction de l'interprétation des séries temporelles, on doit assurer une cohérence entre les scénarios et la traduction des séries symboliques. Pour cela, les scénarios doivent montrer la capacité d'améliorer la traduction symbolique par le biais de la reconnaissance partielle de scénarios. On réalise donc des expérimentations pour évaluer la pertinence des modifications qui peuvent être formulées par le système. Si ces modifications sont pertinentes, on peut penser qu'une fois intégré dans les *agents de reconnaissance de scénarios*, l'algorithme permettra au système de gagner en cohérence.

Apprentissage de scénarios Pour l'apprentissage de scénarios, nous utilisons des données simulées. Plus précisément, nous utilisons le niveau symbolique intermédiaire dans la simulation des données. On génère aléatoirement un scénario et à partir de ce scénario, le simulateur peut générer des séries symboliques qui peuvent servir d'instance (en introduisant des variabilités structurelles et temporelles)

¹⁴Il est à noter qu'il s'agit ici de deux passes différentes de la classification. Mais les résultats étant reproductibles et la corrélation des comportements observés en moyenne est significative.

du scénario. Le test consiste alors à appliquer un apprentissage à partir de ces exemples et de comparer le scénario initial avec le scénario obtenu.

Cette comparaison est réalisée au moyen d'une projection du scénario attendu sur le scénario appris et qui permet d'obtenir une mesure de similarité entre les deux scénarios (*cf.* section C.4 du chapitre 3). La mesure de similarité est entre 0 et 1. Lorsqu'elle est à 0, rien ne correspond entre les deux, lorsqu'elle est à 1 le scénario appris est exactement le scénario attendu (structurellement et temporellement).

Nous avons réalisé deux jeux de tests composés chacun de sept ensembles d'apprentissage contenant 10 exemples d'apprentissage.

- Dans le premier jeu, on fait varier le taux d'erreurs structurelles dans la génération des exemples de scénarios pour savoir si l'algorithme peut tenir compte de variations structurelles dans les instances (*e.g.* absence ou surplus de symboles dans les exemples) et apprendre correctement des scénarios.
- Dans le second jeu, on fait de même en faisant varier la variabilité des informations temporelles pendant la génération de scénarios.

Les jeux de tests nous permettent de juger notre algorithme au regard de ses capacités d'apprentissage à partir de données contenant des erreurs (structurelles ou temporelles) qui peuvent être dues à des erreurs de l'abstraction des séries temporelles en séries symboliques, et de ses capacités à pouvoir corriger ces erreurs.

Dans tous les tests nous avons pris un seuil d'apprentissage de 0.6. C'est-à-dire que les scénarios trouvés devaient être plus que majoritaires dans les exemples. En conséquence, le taux d'erreur structurel doit être toujours en dessous de 0.4 sinon, si on modifie au delà de 40% le scénario original, on sait qu'on aura jamais 60% de scénarios similaires. Le test serait alors sans intérêt. Ce seuil est très élevé par rapport au seuil habituel utilisé pour l'algorithme APriori mais correspond à ce qui est utilisé pour les *agents de construction de scénario*. En effet, ces agents construisent des exemples de l'explication desquels des similarités significatives doivent être trouvés. La significativité se traduit donc ici par la majorité à plus de 60% dans les exemples.

Des expérimentations préliminaires (non présentées ici) ont permis de mettre en évidence que la taille du scénario à apprendre ne modifiait pas significativement les performances de l'algorithme, nous utilisons donc systématiquement des scénarios de taille 4.

Sur le premier jeu de données, les paramètres de simulation des données sont les suivants :

- Taille=4
- NbreSymb=10
- DateMax=15
- VarDate=1
- ErrStruct = 0.05*id (id = identifiant de l'ensemble d'exemples de 1 à 7)

Le résultat de l'apprentissage est illustré ci-dessous (*cf.* figure 4.9). **Explanation** indique le scénario qui a été appris. L'affichage d'un scénario donne la suite des événements qui le constituent. Chaque événement est défini entre parenthèses par un premier chiffre qui identifie l'évènement, puis par l'intervalle [**debut**, **fin**] de l'évènement. **Proto** est le scénario qu'il fallait retrouver. Finalement, **Similarity** donne la mesure de similarité entre l'explication trouvée et le scénario à retrouver. On met en évidence en rouge les différences intéressantes entre les scénarios.

On peut mettre en évidence que le scénario est structurellement quasiment toujours retrouvé. Ce résultat montre le bon fonctionnement de notre algorithme et également l'importance du seuil de l'algorithme APriori. En effet, pour le dernier jeu (ErrStruct=0.35), on ne peut pas retrouver le scénario avec un seuil de 0.6. Seul un sous-scénario a été trouvé suffisamment fréquent.

Sur le second jeu de données, les paramètres de simulation des données sont les suivants :

- Taille=4
- NbreSymb=10
- DateMax=15
- VarDate=0.5*id (id = identifiant de l'ensemble d'exemples de 1 à 7)
- ErrStruct = 0.2

Les résultats de l'apprentissage sont donnés dans la figure 4.10. De nouveau, les structures sont quasiment systématiquement extraites. On ne constate pas de lien évident entre la variance des dates et

```

ErrStruct = 0.05
Explanation : (6, [1.22,3.56]), (9, [2.97,9.70]), (8, [9.01,11.86]), (4, [12.11,14.49])
Proto : (6, [1.27,3.31]), (9, [4.56,7.50]), (8, [9.63,10.65]), (4, [13.28,13.45])
Similarity : 0.435

ErrStruct = 0.1
Explanation : (3, [9.40,12.33]), (8, [12.08,14.10]), (9, [13.74,14.27]), (7, [14.11,15.14])
Proto : (3, [9.98,11.34]), (8, [12.93,13.41]), (9, [13.89,14.03]), (7, [14.08,14.94])
Similarity : 0.437

ErrStruct = 0.15
Explanation : (9, [1.84,3.45]), (7, [3.39,8.44]), (2, [8.05,12.29]), (3, [11.62,15.99])
Proto : (9, [1.88,3.36]), (7, [3.59,8.02]), (2, [8.04,11.49]), (3, [12.59,14.72])
Similarity : 0.771

ErrStruct = 0.2
Explanation : (3, [0.82,3.91]), (6, [8.29,9.92]), (2, [10.36,12.61]), (5, [11.34,15.33])
Proto : (3, [0.86,1.67]), (6, [8.39,9.84]), (2, [11.57,11.62]), (5, [11.95,14.09])
Similarity : 0.428

ErrStruct = 0.25
Explanation : (2, [0.18,1.11]), (8, [1.42,2.90]), (5, [4.65,6.02]), (7, [6.44,14.18])
Proto : (2, [0.62,0.94]), (8, [1.47,2.83]), (5, [5.02,5.51]), (7, [6.50,14.07])
Similarity : 0.649

ErrStruct = 0.3
Explanation : (5, [2.78,5.49]), (4, [4.12,9.07]), (6, [9.07,10.78]), (6, [13.71,14.98])
Proto : (5, [4.11,4.35]), (4, [4.43,8.05]), (6, [9.18,10.49]), (6, [13.87,14.82])
Similarity : 0.581

ErrStruct = 0.35
Explanation : (10, [0.16,4.95]), (3, [3.01,10.54]), (10, [9.22,10.96])
Proto : (10, [2.01,2.93]), (3, [3.17,8.76]), (10, [9.38,9.39]), (9, [13.73,14.38])
Similarity : 0.234

```

FIG. 4.9.

Évaluation de l'apprentissage en fonction de l'erreur structurelle. Le dernier symbole mis en évidence dans la dernière expérimentation montre une erreur dans l'apprentissage de la structure du scénario.

des durées et la mesure de similarité. On aurait pourtant pu s'y attendre : en augmentant les variances des durées et des dates, on augmente les maximums et les minimums des exemples générés, et donc comme l'apprentissage prend en compte les contraintes les plus larges le scénario appris devrait être plus général et ainsi avoir une mesure de similarité inférieure. Or, à part pour le dernier exemple (`VarDate=3.5`), il n'y a pas de chute notable de la mesure de similarité. Si on prend l'avant dernier jeu d'apprentissage (`VarDate=3`), le scénario appris est très similaire à celui qui était attendu. De nouveaux jeux de tests ont été générés et on observe systématiquement que pour `VarDate=3.5` le résultat est au-dessus de la tendance générale à la décroissance. Nous ne sommes à ce jour pas en mesure d'en donner une explication.

Finalement, l'apprentissage de scénario permet de retrouver les scénarios à partir des exemples présentant des variabilités structurelles et temporelles. L'algorithme est donc performant pour générer des explications à partir d'exemples de séries symboliques qui ont des propriétés similaires à celles de nos données simulées.

L'algorithme montre des limites d'expressivité¹⁵. On peut constater que l'apprentissage avec l'algorithme `APriori` extrait les contraintes temporelles les plus larges mais n'extrait pas *réellement* cette information des exemples. Ceci est particulièrement vrai lorsqu'il y a deux occurrences du même événement dans le scénario à apprendre, la construction des contraintes temporelles confond alors les deux occurrences et les deux événements sont associés avec les mêmes contraintes : le début du premier et la fin du second. Contrairement à la segmentation pour laquelle le modèle d'événement limitait l'expressivité, pour l'apprentissage des scénarios, c'est l'algorithme qui limite l'expressivité et non le modèle.

Dans le cadre d'une application à la correction de l'abstraction des séries temporelles, le fait de pouvoir prendre en compte les erreurs structurelles et temporelles nous amène à penser des corrections peuvent être proposées par la méthode de reconnaissance de scénarios.

¹⁵C'est la raison pour laquelle nous avons proposé un algorithme d'apprentissage pouvant apprendre les dimensions temporelles en même temps que la structure du scénario (*cf.* Annexe C).

```

VarDate=0.5
Explanation : (10, [0.31,2.68]), (6, [2.66,12.87]), (8, [14.62,15.12])
Proto : (10, [0.72,1.80]), (6, [3.07,12.43]), (8, [14.78,14.79]), (7, [14.82,14.94])
Similarity : 0.350

VarDate=1
Explanation : (10, [3.31,4.38]), (7, [6.64,7.15]), (4, [7.08,8.61]), (3, [9.57,11.20])
Proto : (10, [3.67,4.18]), (7, [6.74,6.95]), (4, [7.02,7.56]), (3, [9.64,10.99])
Similarity : 0.497

VarDate=1.5
Explanation : (7, [4.69,6.71]), (6, [6.42,7.31]), (8, [9.58,15.56])
Proto : (7, [4.90,6.50]), (6, [6.56,6.61]), (10, [8.87,9.12]), (8, [11.08,12.66])
Similarity : 0.277

VarDate=2
Explanation : (7, [1.42,6.41]), (7, [5.90,7.81]), (10, [8.21,12.31]), (5, [11.78,15.63])
Proto : (7, [1.75,6.19]), (7, [6.45,6.46]), (10, [8.98,11.80]), (5, [12.43,14.20])
Similarity : 0.511

VarDate=2.5
Explanation : (7, [2.60,5.97]), (4, [5.14,12.60]), (5, [10.14,14.62])
Proto : (7, [2.95,4.61]), (4, [6.14,10.72]), (5, [11.75,12.29]), (10, [14.36,14.60])
Similarity : 0.306

VarDate=3
Explanation : (10, [0.37,3.85]), (10, [2.89,5.33]), (10, [3.84,13.42]), (6, [14.11,14.98])
Proto : (10, [0.91,1.79]), (10, [3.98,4.16]), (10, [4.77,10.14]), (6, [14.25,14.90])
Similarity : 0.412

VarDate=3.5
Explanation : (9, [1.69,5.65]), (3, [6.507,10.58]), (10, [8.69,11.95]), (3, [10.09,14.71])
Proto : (9, [3.45,4.40]), (3, [7.87,8.23]), (10, [8.41,9.63]), (3, [12.39,13.61])
Similarity : 0.198

```

FIG. 4.10.
Évaluation de l'apprentissage en fonction de la variabilité temporelle des événements.

Reconnaissance de scénarios On s'intéresse maintenant au résultat d'une reconnaissance partielle de scénario, c'est-à-dire la proposition d'une modification d'un exemple dans le but de corriger une mauvaise symbolisation sur la base d'une amélioration de la confiance d'un scénario.

On donne dans la figure 4.11, les propositions de modifications pour un des 8 jeux d'apprentissage contenant des erreurs structurelles importantes pour favoriser les opportunités d'apporter des modifications intéressantes. Lors de l'apprentissage d'une explication, l'algorithme a aussi généré les motifs le plus intéressants à retrouver pour améliorer les séries symboliques (Sur l'exemple il y a deux `MotifToChange`). La méthode de reconnaissance de scénario parcourt les exemples et propose alors des modifications sous la forme de l'insertion d'un symbole dans l'instance. On peut noter que l'introduction proposée correspond bien à l'absence d'un symbole dans le motif initial d'un exemple (absence due à une perturbation structurelle du scénario lors de la simulation). La proposition est systématiquement répercutée sur nos exemples et nous réalisons de nouveau l'apprentissage sur le jeu de données modifié. Nous comparons ensuite les résultats obtenus avec les scénarios attendus.

Maintenant que nous sommes assurés du bon fonctionnement les modifications introduites dans les séries symboliques, nous réalisons une évaluation pour mettre en évidence l'amélioration qui peut être apportée à l'apprentissage par les modifications. En effet, les modifications peuvent être dite efficaces lorsqu'elles permettent d'améliorer l'apprentissage des motifs. Nous procédons pour cela à une évaluation de l'amélioration de la mesure de similarité en exécutant un cycle apprentissage, reconnaissance avec introduction des corrections puis de nouveau un apprentissage. Nous comparons ensuite les similarités des explications obtenues par les deux apprentissages successifs.

Sur la figure 4.12, on donne les résultats pour le second jeu de données (avec des erreurs temporelles variables). Cette exemple illustre les différentes situations qui peuvent être observées :

- Amélioration de la similarité par la complétion de l'apprentissage. C'est le cas pour `VarDate=0.5` et `1.5`. Les modifications introduites permettent de trouver une explication plus complète au second apprentissage
- Diminution de la similarité par l'élargissement des contraintes temporelles. Les motifs introduits

Chapitre 4. Évaluations

```
***** Explication *****
motif :(6, [3.74,9.20]), (1, [7.09,9.47]), (4, [9.41,14.63]), (3, [14.42,16.24])

***** MotifToChange *****
motif :(6, [3.74,9.20]), (1, [7.09,9.47]), (4, [9.41,14.63])

***** MotifToChange *****
motif :(6, [3.74,9.20]), (1, [7.09,9.47]), (3, [14.42,16.24])
Modification :
Du motif : (6, [4.11,6.38]), (9, [6.51,6.51]), (1, [6.52,6.99]), (2, [9.46,9.76]), (4, [11.67,13.76]),
(6, [14.27,14.38])
Intro de : (3 [14.42 16.24])
Modification :
Du motif : (6, [5.16,7.74]), (1, [7.74,8.25]), (3, [11.28,11.45]), (9, [11.57,13.90]), (8, [15.35,-
15.36])
Intro de : (4 [9.41 14.63])
Modification :
Du motif : (6, [5.64,8.25]), (1, [8.25,8.75]), (2, [9.32,9.54]), (3, [13.76,13.82])
Intro de : (4 [9.41 14.63])
```

FIG. 4.11.

Exemple de génération d'une modification à partir de la reconnaissance partielle d'un scénario. L'illustration donne tout d'abord l'explication apprise puis les motifs à retrouver qui peuvent amener des améliorations. Enfin, trois modifications sont proposées : sur trois des exemples d'apprentissage l'algorithme propose d'insérer un symbole à chaque fois. Sur les exemples, on indique en rouge les symboles qui sont déjà dans l'explication.

le sont avec les contraintes temporelles apprises (les plus larges), en conséquence lors d'un second apprentissage les contraintes temporelles peuvent être élargies et diminuer ainsi la similarité. Par exemple, pour le jeu d'exemples avec `VarDate=2` et `3.5`. C'est également ce qui se passe pour `VarDate=2.5`, malgré l'insertion d'un nouveau motif qui améliore la structure de l'explication apprise, les contraintes temporelles ont également été modifiées.

- Amélioration de la similarité par l'élargissement des contraintes temporelles. C'est un cas de la situation où `VarDate=3`. Cet élargissement des contraintes correspond à une difficulté de faire des modifications correctes lorsque plusieurs évènements identiques se succèdent.

Sur 21 exemples, 4 améliorations ont été apportées aux scénarios, 10 diminutions de la similarité ont été observées. Les autres explications n'ont pas été modifiées par l'introduction de modifications dans les exemples. En moyenne, les diminutions de la similarité sont de 6% tandis que les améliorations sont de 13%. On n'identifie aucune amélioration de la similarité due à une modification des contraintes temporelles (hors du cas `VarDate=2.5`). Ceci est dû au fait qu'on insère uniquement des symboles avec les contraintes temporelles les plus larges. À l'inverse, on observe aucune diminution de la similarité par une modification erronée de la structure d'une explication.

Si on revient au cas d'un scénario avec deux occurrences du même évènement évoqué pour l'apprentissage de scénario. Le fait d'apprendre un scénario erroné a pour conséquence l'introduction erronée de cet évènement par le système si il trouve une modification à faire. L'erreur étant algorithmique, ce biais sera permanent dans le système et on ne peut pas compter sur une amélioration par la dynamique globale du système. Au mieux, le biais conduira le système à construire des scénarios contenant uniquement des évènements différents, au pire il peut déstabiliser toute l'interprétation. Dans la mesure où nous avons proposé un algorithme pour résoudre cette difficulté (*cf.* Annexe C), nous n'avons pas exploré plus à fond les limites de notre implémentation. Beaucoup d'expérimentation serait à mener, nous le considérons donc comme une perspective de ce travail.

Le bilan des évaluations de la méthode d'amélioration de la symbolisation par la reconnaissance partielle des explications construites est donc équilibré : d'une part les modifications introduites permettent d'améliorer dans certains cas la structure des explications sans la détériorer, mais d'autre part la correction par l'introduction de symboles avec les contraintes temporelles les plus larges conduit à des imprécisions des contraintes temporelles dans l'apprentissage suivant.

Il est difficile à ce stade de savoir si les corrections erronées réalisées par les *agents de reconnaissance*

VarDate=0.5
 Proto : (1,10, [0.72,1.80]), (1,6, [3.07,12.44]), (1,8, [14.78,14.80]), (1,7, [14.83,14.95])
 Explanation : (1,10, [0.31,2.68]), (1,6, [2.66,12.87]), (1,8, [14.62,15.12])
 Similarity : 0.350
 Explanation : (1,10, [0.31,2.68]), (1,6, [2.84,12.87]), (1,8, [14.62,15.12]), (1,7, [14.77,15.30])
 Similarity : 0.411

VarDate=1.5
 Proto : (1,7, [4.90,6.51]), (1,6, [6.57,6.61]), (1,10, [8.88,9.12]), (1,8, [11.08,12.66])
 Explanation : (1,7, [4.69,6.71]), (1,6, [6.42,7.31]), (1,8, [9.59,15.56])
 Similarity : 0.277
 Explanation : (1,7, [4.69,6.71]), (1,6, [6.39,7.31]), (1,10, [7.79,10.19]), (1,8, [9.59,15.56])
 Similarity : 0.302

VarDate=2.0
 Proto : (1,7, [1.75,6.19]), (1,7, [6.45,6.47]), (1,10, [8.98,11.80]), (1,5, [12.44,14.21])
 Explanation : (1,7, [1.43,6.41]), (1,7, [5.91,7.81]), (1,10, [8.22,12.31]), (1,5, [11.78,15.64])
 Similarity : 0.527
 Explanation : (1,7, [1.43,6.41]), (1,7, [5.91,7.81]), (1,10, [8.22,12.31]), (1,5, [11.39,15.64])
 Similarity : 0.517

VarDate=2.5
 Proto : (1,7, [2.95,4.61]), (1,4, [6.15,10.72]), (1,5, [11.75,12.30]), (1,10, [14.36,14.61])
 Explanation : (1,7, [2.60,5.97]), (1,4, [5.14,12.60]), (1,5, [10.14,14.63])
 Similarity : 0.306
 Explanation : (1,7, [0.35,5.74]), (1,4, [5.14,12.60]), (1,5, [10.14,14.63]), (1,10, [12.65,15.85])
 Similarity : 0.279

VarDate=3
 Proto : (1,10, [0.92,1.80]), (1,10, [3.99,4.16]), (1,10, [4.77,10.14]), (1,6, [14.26,14.90])
 Explanation : (1,10, [0.37,3.86]), (1,10, [2.90,5.34]), (1,10, [3.84,13.43]), (1,6, [14.11,14.98])
 Similarity : 0.438
 Explanation : (1,10, [0.37,3.86]), (1,10, [2.90,13.43]), (1,10, [3.84,13.43]), (1,6, [14.11,14.98])
 Similarity : 0.525

VarDate=3.5
 Proto : (1,9, [3.45,4.41]), (1,3, [7.88,8.23]), (1,10, [8.41,9.64]), (1,3, [12.39,13.62])
 Explanation : (1,9, [1.70,5.66]), (1,3, [6.51,10.59]), (1,10, [8.69,11.95]), (1,3, [10.10,14.72])
 Similarity : 0.220
 Explanation : (1,9, [1.70,5.66]), (1,3, [6.51,10.59]), (1,10, [6.87,11.95]), (1,3, [10.10,14.72])
 Similarity : 0.208

FIG. 4.12.

Illustration des résultats sur le jeu de données avec erreurs temporelles variables. Proto donne le scénario à apprendre.

de scénarios vont limiter la dynamique globale du système. Dans la section suivante, nous nous sommes donc intéressé à mieux qualifier la dynamique du système dans sa globalité.

B.2 Expérimentations de la cohérence des interprétations construites

B.2.1 Expérimentation de la convergence sur données simulées

L'objectif de l'expérimentation sur des données simulées est d'évaluer la capacité du système à produire une interprétation cohérente des données. D'une part, on cherche à savoir si le système peut converger vers un état stable de l'interprétation et d'autre part on cherche à montrer que le système permet d'adapter son interprétation au cours de son "expérience".

Nous disposons de quatre jeux de données simulées différents et présentant des difficultés semblables. Chaque jeu est composé de trois séries temporelles univariées. Il est généré à partir de quatre événements et d'un scénario à découvrir. Chaque événement est caractérisé par une trace d'une forme particulière. Pour chaque jeu les formes utilisées sont également différentes. Finalement, chaque série temporelle d'un même jeu contient une instance du scénario au milieu d'événements qui ont été rajoutés aléatoirement sur le reste de la série.

À partir de jeux de données simulés, on annote une des séries temporelles uniquement par des segments. Les segments sont positionnés de sorte à correspondre à la segmentation attendue. Les deux autres séries temporelles du jeu sont laissées vierges d'annotations. Le traitement des agents permettant la symbolisation des séries temporelles est alors lancé. L'apprentissage et reconnaissance de scénarios sont laissés de côté pour cette expérimentation. Chaque agent réalise son cycle de traitement une seule fois. Et on répète successivement cette opération 14 fois¹⁶ de sorte à obtenir l'évolution pas à pas de l'interprétation réalisée par le système. À chaque pas, on calcule la mesure de similarité avec le résultat attendu. Ici, l'interprétation désigne uniquement la symbolisation de la série temporelle, la similarité est donc calculée entre la symbolisation produite et la symbolisation attendue.

B.2.2 Résultats

Le résultat de l'expérimentation montre que les mécanismes mis en place permettent au système :

- de reproduire des annotations d'événements donnés *a priori* lorsqu'ils sont amples, et donc d'assurer en partie une cohérence avec l'interprétation de l'utilisateur,
- de produire des éléments d'interprétation qui lui son propre,
- d'arriver à une *forme de stabilisation* de son interprétation.

La *forme de stabilisation* qui peut être observée est celle d'un système qui montre la capacité de reproduire des comportements d'un pas à l'autre (*i.e.* d'un point de vu global : une partie des segments annotés par le même événement se retrouvent annotés de la même manière au cours des annotations successives. Cela donne une impression de reproductibilité des annotations), mais la stabilisation de l'interprétation telle qu'elle a été définie dans la section A.1.1 n'est pas atteinte.

Avant de donner les résultats obtenus pour l'ensemble du protocole d'évolution de la similarité, on illustre les observations faites sur le comportement du système dans un cas particulier (Utilisation du Jeu 3). Ces premières observations nous permettrons d'interpréter les résultats globaux et en particulier d'expliquer les raisons de la non-convergence de la mesure de similarité et pourquoi cette mesure décroît au cours de l'interprétation.

Les figures 4.13 à 4.19 illustrent l'évolution pas à pas de l'annotation d'une série¹⁷. La figure 4.18 donne l'évolution des mesures de similarité. Au travers de cette évolution, on peut suivre l'interprétation réalisée par le système.

L'interprétation initiale (*cf.* figure 4.13) a été obtenue par le premier pas de traitement des agents. Ces traitements ont été possibles grâce aux segments initialement introduits (non visualisés). Globalement, l'interprétation réalisée est plutôt bonne. On peut constater que :

¹⁶Le choix de 14 pas est empirique. Le nombre de pas minimum à faire est de trois. Une modification peut avoir a des répercutions au minimum sur les deux pas suivant. Nous avons commencé par réaliser un plus grand nombre de pas, mais au delà de 14, on n'observe plus de comportements différents de ceux qui seront relatés par la suite.

¹⁷On suit l'évolution de l'annotation de la série qui a été initialement annotée par des segments. Il est important de noter que l'interprétation est réalisée en même temps sur deux autres signaux dont les annotations influencent également l'annotation observée. L'illustration ne retranscrit donc pas toute l'interprétation du système.

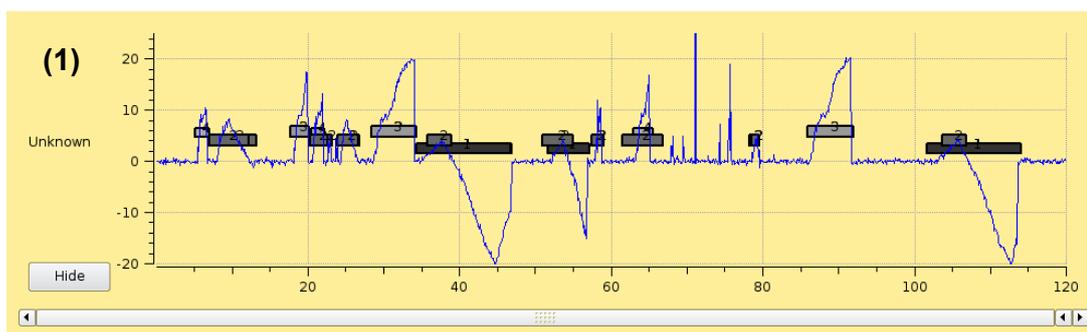


FIG. 4.13.
Évolution de l'annotation d'une série temporelle : Étape 1. (Jeu de données 3, série annotée 2).

- Les formes de taille importantes sont bien reconnues (symboles 1 et 3) ;
- Le symbole 4 pour des pics de tailles inférieures est plus difficilement retrouvé (à la date 20, il y a une ambiguïté entre le 3 et le 4) ;
- Le symbole 2 qui a pour profil une sorte de bosse à un double emploi qui ne se trouve initialement pas dans les données : il est utilisé pour les segments qui on une forme de bosse, mais se retrouve également à l'intérieur des motifs annotés par le symbole 1.

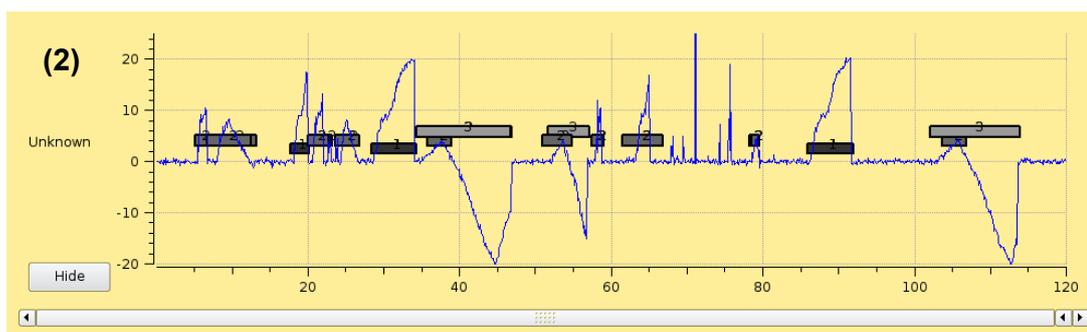


FIG. 4.14.
Évolution de l'annotation d'une série temporelle : Étape 2. On observe des modifications d'attribution des numéros de symboles.

Pour la seconde étape (*cf.* figure 4.14), les nouveaux symboles qui ont été introduits par le système sont réintroduits dans le traitement. Au cours de ce second traitement, on se rend compte que le système a changé l'attribution des identifiants de symboles. Le symbole 4 a disparu, le symbole 1 est devenu 3 et 3 est devenu 1. Ces changements d'identifiants de symboles ne sont pas considérés comme des différences lors du calcul de la similarité. Globalement, on constate assez peu de modifications de l'interprétation.

Lors de la troisième étape (*cf.* figure 4.15), on constate la réapparition du symbole 4 annotant quasiment les mêmes segments que lors de l'étape 1. Les deux pics autour de la position 20 sont alors considérés comme identiques. À l'étape 4, ces deux pics sont de nouveau distingués revenant ainsi à une interprétation très proche de celle de l'étape 1. Au travers de l'analyse de ces premières étapes, on peut constater un phénomène d'"hésitation" sur certaines interprétations. Lorsque deux segments sont assez proches, ils sont alternativement considérés comme identiques puis différents. Ce bouclage d'ordre 2 est une première limite à la convergence de l'interprétation. Ces bouclages peuvent parfois être dépassés par des modifications extérieures.

La transition entre l'étape 4 et l'étape 5 (*cf.* figure 4.16) montre une sorte de *transition de phase* : Sur la figure 4.18, on observe une chute brutale de la similarité entre deux étapes : il y a eu de gros changements dans l'interprétation. Les transitions de phase apparaissent ainsi de manière non attendue. Le symbole 2 a maintenant une caractérisation beaucoup plus floue : il est utilisé à la fois pour annoté des portions plates, des bosses (en 10) et même des pics (en 5). Un tel changement peut être vu comme une manifestation objective d'un phénomène émergent. Il faut noter qu'il y a toujours un décalage entre l'annotation et les modèles puisque les modèles ont été ici construit à partir des annotations de l'étape

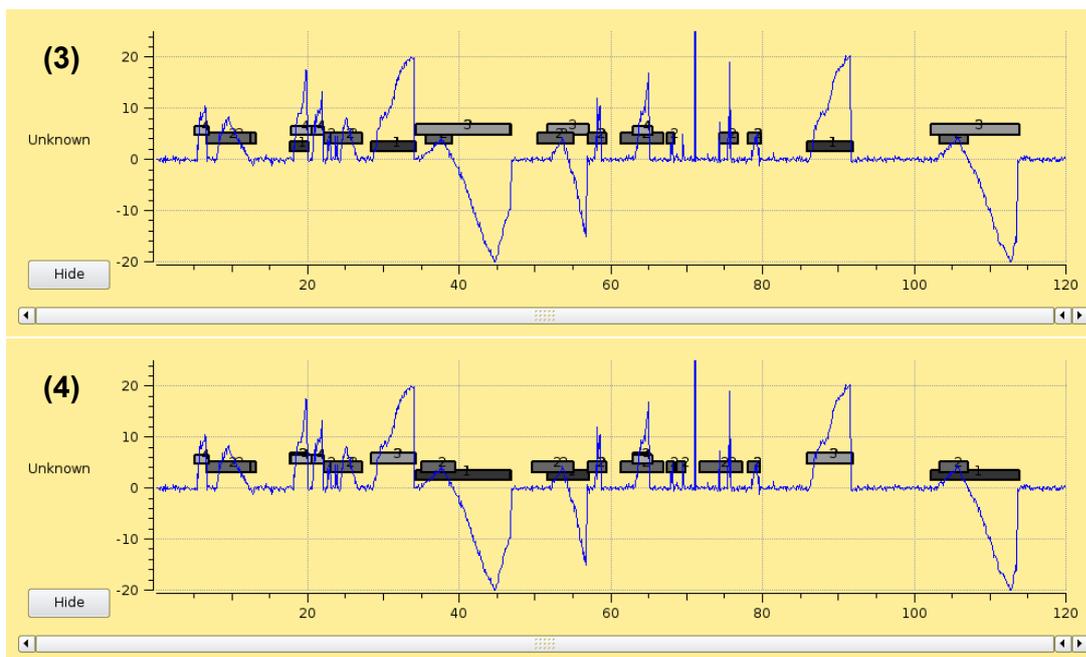


FIG. 4.15. Évolution de l'annotation d'une série temporelle : Étapes 3 et 4. *Hésitation* sur les annotations des symboles 3 et 4.

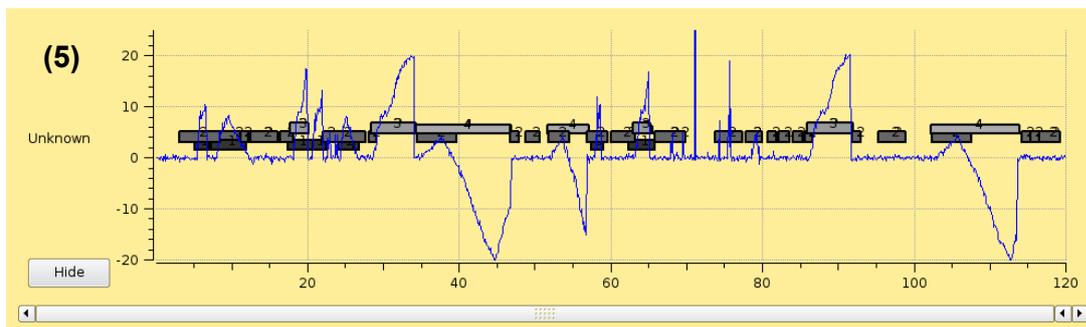


FIG. 4.16. Évolution de l'annotation d'une série temporelle : Étape 5. Observation d'une transition de phase : de nouveaux événements ont émergés.

4. Les conséquences pour la mesure de similarité avec l'annotation attendue, qui était jusque là plutôt bonne (cf. figure 4.18, apparaît dans les étapes suivantes avec une réduction générale¹⁸ de la similarité.

Dans les étapes suivantes (n°6, 7, 8) (cf. figure 4.17), les choses se clarifient un peu, on peut identifier une mise en cohérence entre les annotations et les modèles par une augmentation de la différence entre les interprétations successives. Sur la figure 4.18, on observe une croissance de la courbe.

À l'étape 6, le système introduit une différenciation entre les segments 2 : De nouveaux types de symboles (les 3, 4) apparaissent pour représenter les aplats, distingués des bosses et des pics. On peut noter que les types de symboles 4 et 5 restent relativement stables. Les formes amples sont suffisamment bien caractérisées pour être stables.

Le symbole 3 à la position 92, en bordure d'une chute brutale, est intéressant pour la suite. Dans l'étape 7 suivante, ce symbole est différencié des autres 3 (alors indiqués par un 5). Ce nouveau type de forme est pérennisé dans la suite des interprétations. On illustre ainsi le mécanisme par lequel de nouvelles formes peuvent apparaître au système : Par une annotation un peu perturbée, de nouveaux segments sont mis en jeu dans le processus et trouvent d'eux-mêmes un intérêt à exister et se pérenniser. Ce type de symbole est une illustration de la capacité du système à produire ces propres éléments d'interprétation.

¹⁸la similarité est calculée à partir des trois séries temporelles interprétées

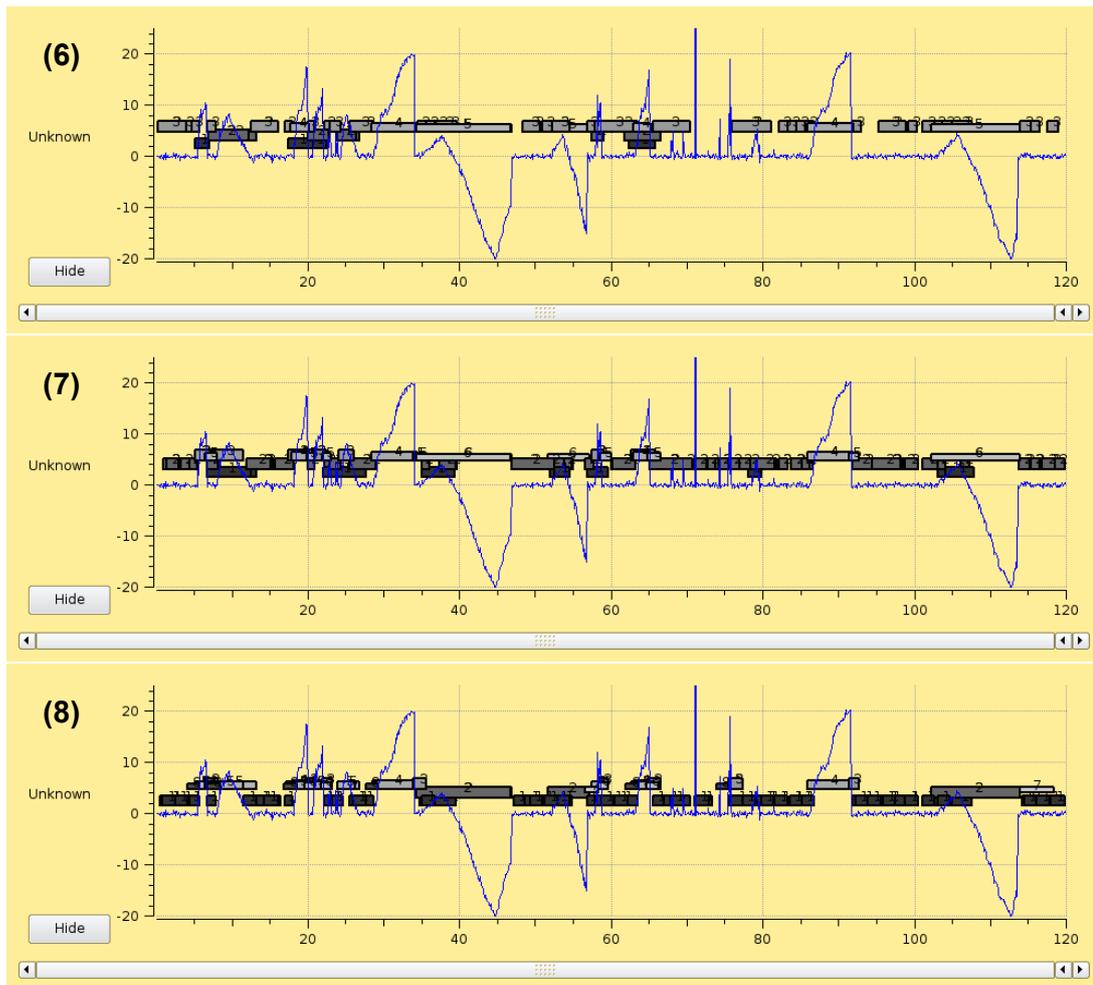


FIG. 4.17.
Évolution de l'annotation d'une série temporelle : Étapes 6, 7 et 8. Confirmation des nouveaux symboles.

Dans les étapes suivantes (n°8 à 14) (*cf.* figure 4.19) on constate une stabilisation générale de l'interprétation. Mais nous observons cette stabilisation en tant qu'observateur extérieur capable de faire des généralisations. Pour le système, il est difficile de mettre en évidence cette stabilisation. Même si on observe la stabilisation de la mesure de similarité entre interprétations successives, nous avons vu qu'il peut y avoir des modifications brutales.

Au final¹⁹ (à l'étape 14), l'interprétation qui a été réalisée par le système est assez loin (au sens de la mesure utilisée) de l'interprétation attendue initialement. Cependant, elle ne semble pas mauvaise pour autant : elle présente une certaine cohérence. Une fois comprise, cette interprétation peut s'argumenter. Les types de symboles 4 et 7 permettent d'identifier les formes amples. Les types de symboles 3 et 6 ont été découverts par le système, l'un pour les aplats et l'autre pour les chutes successives. Il est intéressant de noter que ces deux formes sont retrouvées de manière récurrente dans des jeux de données différents et sur des essais différents. Ces formes apparaissent ainsi comme des éléments très saillants pour notre implémentation du système. Le type de symbole 2 correspond à des bosses de courtes durées et le 1 à une montée puis une rechute.

Nous reportons maintenant les résultats de l'évolution pas à pas de la similarité entre l'annotation réalisée et l'annotation attendue dans les figures 4.20, 4.21, 4.22 et 4.23. Les annotations initiales et finales des jeux de données 1, 2 et 4 sont également données pour une des séries temporelles du jeu dans les figures 4.24, 4.25, 4.26. Chacune des trois courbes d'une figure est obtenue comme la moyenne des

¹⁹ Il s'agit de l'étape finale pour l'expérimentation, mais l'interprétation par le système n'est pas finie.

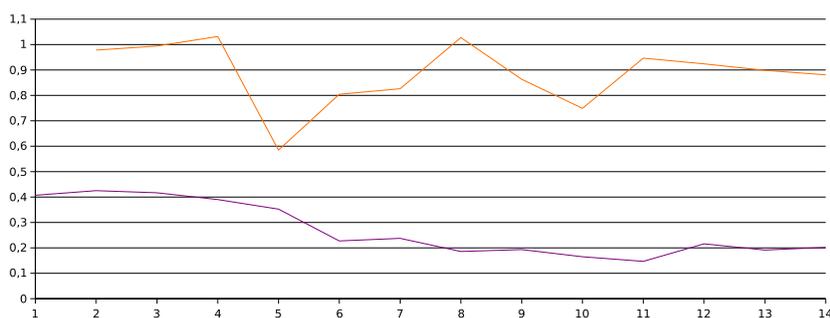


FIG. 4.18.

La courbe supérieure donne l'évolution de la similarité entre interprétations successives de l'interprétation du jeu 3. La courbe inférieure donne la similarité entre l'annotation courante et l'annotation attendue.

similarités de l'annotation de chaque série avec l'annotation attendue.

La première constatation qui peut être faite à partir de ces figures est que l'annotation symbolique des séries temporelles ne se stabilise pas sinon, la mesure elle-même se stabiliserait.

Pour le jeu 3, on constate pour les trois expérimentations menées qu'il y a initialement une chute importante de la mesure de similarité, et ensuite une stabilisation. Nous avons expliqué précédemment cette chute comme un *changement de phase* par l'introduction de formes propres au système qui distinguent ainsi l'annotation qu'elles réalisent de celle qui est attendue.

Pour le jeu 4, la mesure de similarité reste à un niveau élevé (≈ 0.5). Dans ce jeu de données, les formes mises en jeu dans l'annotation sont principalement des formes amples et dont la distribution dans le temps les sépare bien. Par conséquent, la reconnaissance faisant moins d'erreurs (similarité initiale élevée), il n'y a pas de *changement de phase* observé.

Pour le jeu 1, la mesure de similarité commence également à un niveau élevé (≈ 0.55) mais on peut tout de même identifier des transitions de phase. Il est intéressant de constater que dans les trois interprétations de ce jeu (avec des annotations initiales différentes), la transition de phase n'apparaît pas au même pas d'exécution contrairement au jeu 4 pour lequel les trois courbes chutes au début. Pour la courbe inférieure, il n'y a pas de transition de phase réellement identifiée mais plutôt une lente dégradation. Pour la courbe médiane, la chute apparaît au pas 5, et pour la courbe supérieure, elle est retardée au pas 11. L'apparition d'un tel phénomène est donc bien difficilement prévisible.

Finalement, le jeu 2 présente une baisse initiale relativement importante puis une stabilisation globale avec une très légère tendance à la hausse de la similarité. La similarité initiale est faible (≈ 0.3). On se rend compte sur la figure 4.25 que dès le début les aplats ont été détectés réduisant ainsi la similarité.

B.2.3 Apport de la reconnaissance de scénarios à la convergence de l'interprétation

En suivant le même protocole, nous avons rajouté le traitement des agents qui réalisent l'apprentissage et la reconnaissance des scénarios. L'objectif de cette expérimentation est de voir si l'utilisation de scénarios, avec l'introduction d'un *feedback* sur la symbolisation des séries temporelles permet d'améliorer le traitement global.

La figure 4.27 montre l'évolution pas à pas de la mesure de similarité pour les jeux 3 et 4. La figure 4.28 montre le résultat obtenu à la fin des 14 pas pour le jeu 4²⁰. On voit sur l'évolution de la similarité pas à pas du jeu 3 (*cf.* figure 4.27) que l'utilisation de l'apprentissage de scénarios ne permet pas d'empêcher les phénomènes de transition de phase. Mais on constate que les courbes pour les jeux 3 et 4 sont plus lisses. L'utilisation du feedback permet de réguler le comportement du système. Les résultats sur les similarités avec l'annotation attendue sont du même ordre que pour les expérimentations sans scénarios. Cependant, on peut noter qu'en moyenne, l'utilisation des scénarios limite la réduction de la similarité. En moyenne, la similarité passe de 0.43 à 0.32 lors de l'exécution de 14 pas par le système sans utilisation des scénarios, alors que la similarité passe simplement de 0.44 à 0.40 avec l'utilisation des scénarios. Donc la régulation effectuée par les scénarios permet de maintenir une cohérence plus forte avec les annotations initiales.

²⁰Pour les autres jeux de données, il est difficile de visualiser le résultat car l'abondance de symboles conduit également à une abondance de scènes.

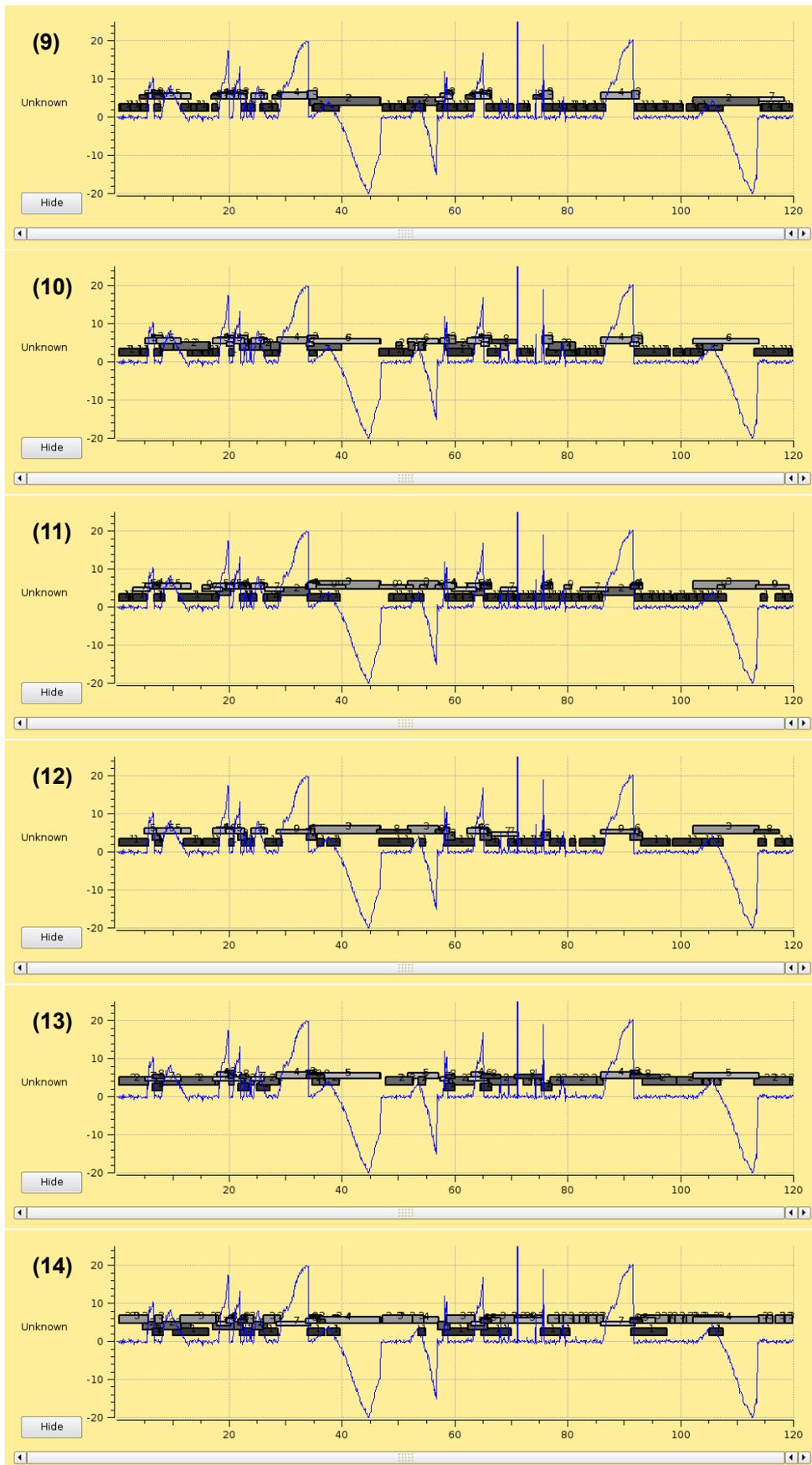


FIG. 4.19. Finalisation de l'évolution de l'interprétation pas à pas (9 à 14).

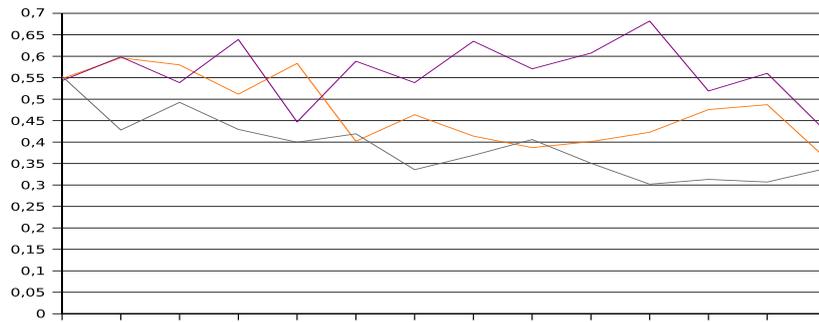


FIG. 4.20. Évolutions de la similarité dans une interprétation pas à pas : Jeu de données 1. Chaque courbe correspond à l'une des séries du jeu qui a servi des annotations d'origine.

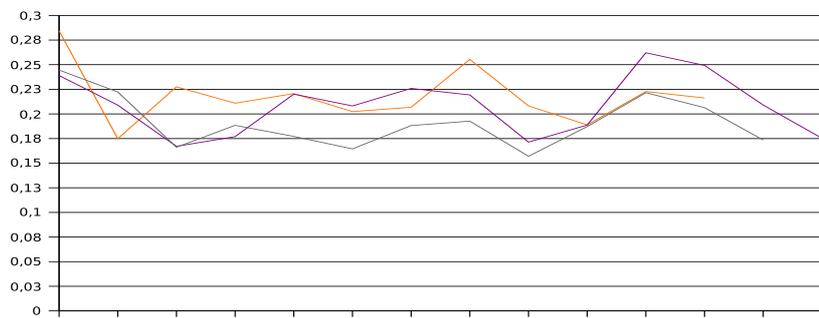


FIG. 4.21. Évolutions de la similarité dans une interprétation pas à pas : Jeu de données 2. Chaque courbe correspond à l'une des séries du jeu qui a servi pour les annotations d'origine.



FIG. 4.22. Évolutions de la similarité dans une interprétation pas à pas : Jeu de données 3. Chaque courbe correspond à l'une des séries du jeu qui a servi pour les annotations d'origine.

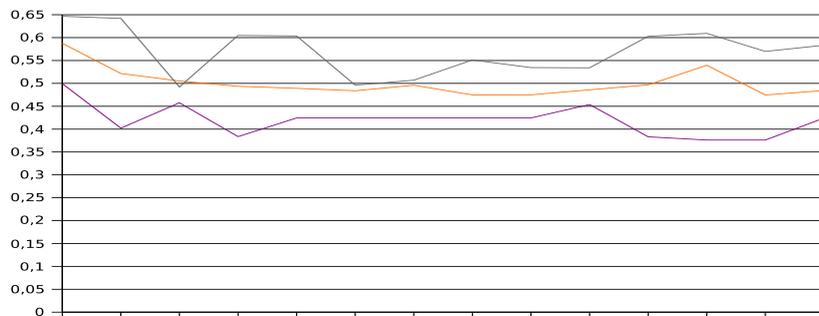


FIG. 4.23. Évolutions de la similarité dans une interprétation pas à pas : Jeu de données 4. Chaque courbe correspond à l'une des séries du jeu qui a servi pour les annotations d'origine.

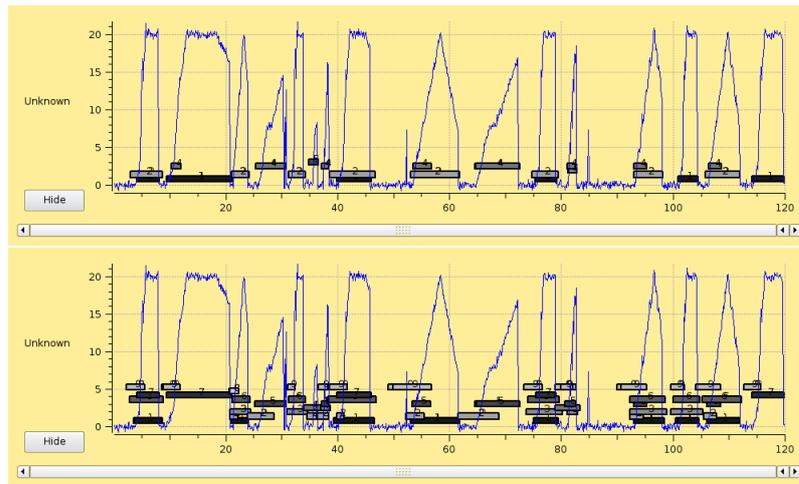


FIG. 4.24. Annotation initiale (au dessus) et finale (en dessous) d'une série temporelle du jeu 1.

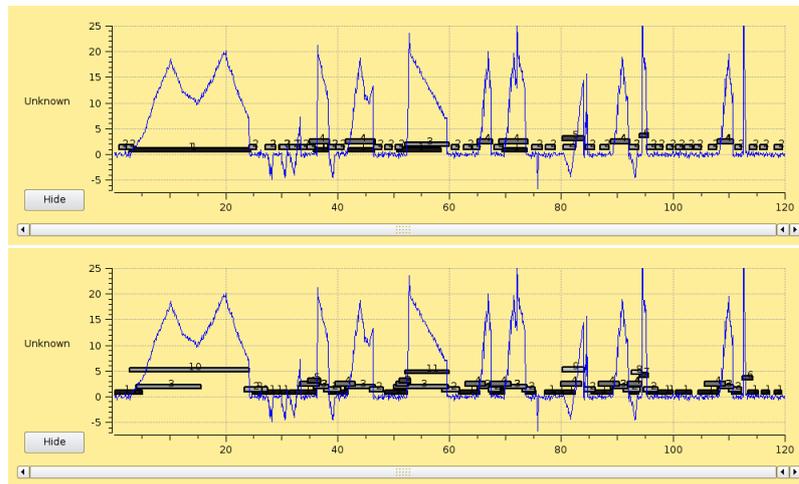


FIG. 4.25. Annotation initiale (au dessus) et finale (en dessous) d'une série temporelle du jeu 2.

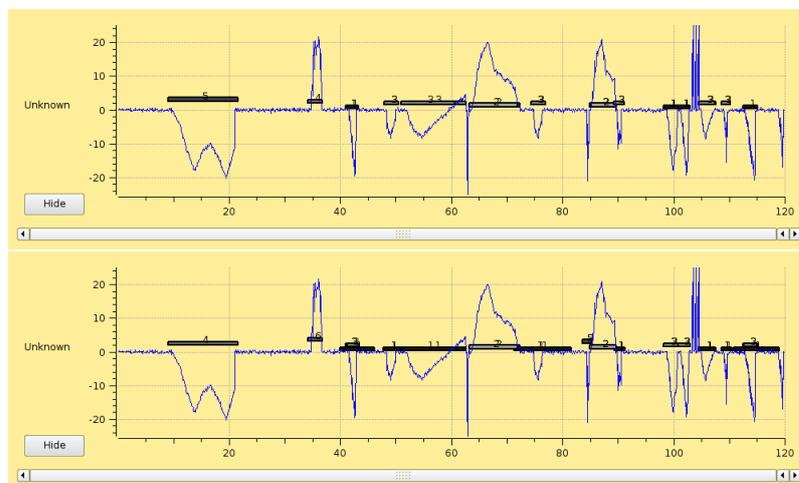


FIG. 4.26. Annotation initiale (au dessus) et finale (en dessous) d'une série temporelle du jeu 4.

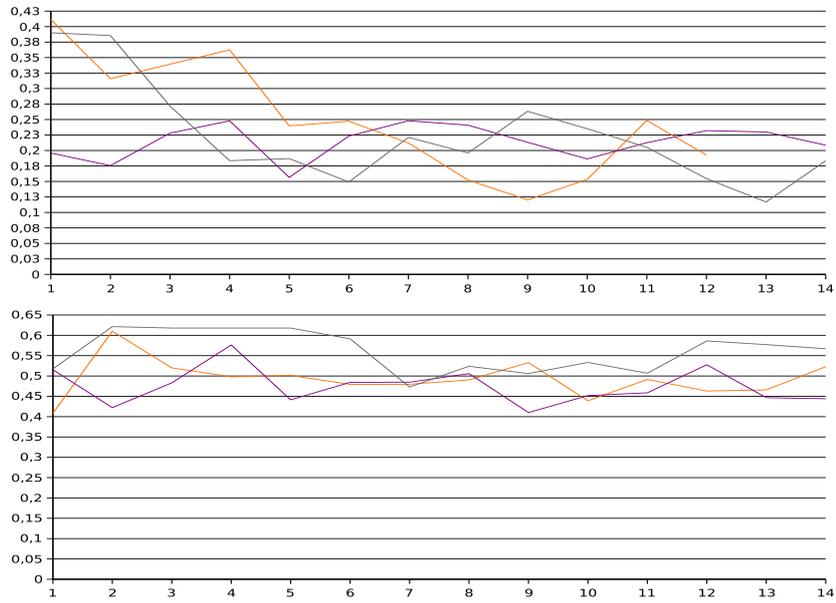


FIG. 4.27. Évolutions de la similarité dans une interprétation pas à pas avec scénarios : au dessus, le jeu 3 et en dessous le jeu 4.

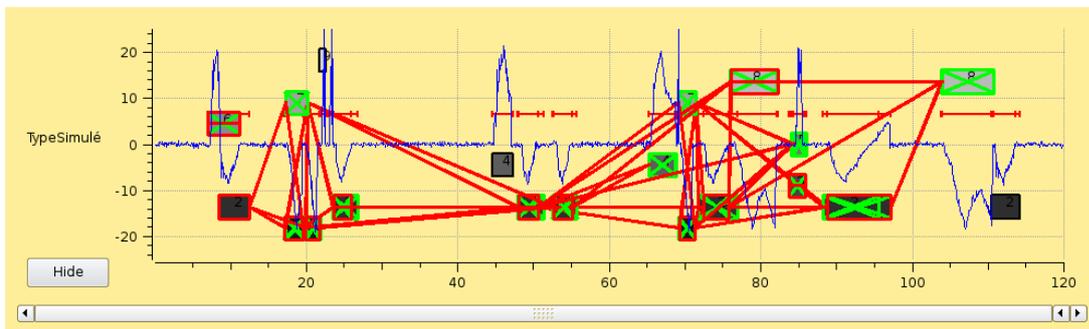


FIG. 4.28. Annotation finale d'une série temporelle du jeu 4 avec scènes.

C Évaluation de la pertinence de la collaboration

L'évaluation de la collaboration a pour but d'évaluer les qualités et défauts de notre approche de la collaboration par rapport à d'autres approches de la tâche d'interprétation des séries temporelles. On fait donc une étude comparative de différents modes d'interprétation de séries temporelles.

C.1 Expérimentations

Dans cette expérimentation du système, on implique des utilisateurs pour évaluer différents modes d'interprétation des séries temporelles. On présente initialement aux utilisateurs la tâche qu'ils ont à accomplir, à savoir l'annotation et l'interprétation des trois séries temporelles (univariées) à l'aide de *segments*, *symboles* et *scènes*. Ces notions leur sont alors expliquées.

Un utilisateur utilise quatre modes d'interprétation de séries temporelles :

- Mode manuel : Les utilisateurs disposent de séries temporelles imprimées sur papier et réalise une annotation et une interprétation uniquement sur le support papier.
- Mode informatisé (Info) : L'utilisateur réalise l'annotation et l'interprétation des séries temporelles.
- Mode automatique contrôlé (Cont) : L'utilisateur peut disposer des fonctionnalités de traitement des agents du système à la demande : segmentation, symbolisation, apprentissage et reconnaissance de scénarios. "A la demande" signifie ici que lorsque l'utilisateur souhaite une intervention d'agents, ceux-ci réalisent leur traitement une unique fois sur l'ensemble des trois séries temporelles.
- Mode automatique (Auto) : L'utilisateur utilise un système qui annote les séries temporelles en même temps que lui. Notre prototype ne permettant pas d'assurer pleinement cette fonctionnalité, ce mode est simulé par une technique de Magicien d'Oz [7]. La technique de Magicien d'Oz consiste à simuler le comportement d'un système lorsque celui-ci n'est pas totalement implémenté sans rien en dire à l'utilisateur expérimenté qui croit utiliser un système informatisé²¹. L'expérimentateur dispose d'un ordinateur sur lequel il visualise l'annotation réalisée par l'utilisateur et peut de son côté modifier et ajouter des annotations que l'utilisateur prend pour des interventions du système. Les transmissions des informations entre l'ordinateur de l'expérimentateur et l'ordinateur de l'expérimenté(e) se fait au travers du réseau local. Ces interventions sont prévues initialement avec la connaissance des données et du fonctionnement du système simulé. L'expérimentateur doit s'approcher au plus près du fonctionnement attendu de sorte à ce que le retour d'utilisation donné par l'utilisateur soit en accord avec ce que l'on souhaite expérimenter.

Les quatre modes sont successivement testés dans l'ordre de la liste précédente. Cet ordre est systématiquement le même. Ceci permet à l'utilisateur d'appréhender progressivement les fonctionnalités et l'utilisation de l'outil informatique. La description des possibilités des différents modes ainsi que leur utilisation est décrite au préalable de chaque utilisation.

Pour les séries temporelles nous utilisons les quatre jeux de données simulées de l'évaluation de la faisabilité du système (*cf.* section B.2). Pour chaque jeu, les formes à découvrir (pour identifier les événements) sont différentes de sorte à ce qu'à chaque mode testé, l'utilisateur soit face à de nouvelles formes à interpréter. Pour chaque utilisateur on attribue aléatoirement le jeu de données qui est utilisé pour un mode.

Pour chaque mode, l'utilisateur dispose de 5 minutes pour identifier les différents événements (dont il ne connaît pas le nombre) et le scénario (qu'il sait présent dans les séries).

L'évaluation est faite sur des critères objectifs et des critères subjectifs. L'annotation finale est enregistrée et comme les données ont été simulées, nous sommes en mesure de comparer de manière objective la réalisation avec l'annotation attendue. De plus, pendant les expérimentations informatisées, nous traçons toutes les actions réalisées par l'utilisateur de sorte à décompter :

- le nombre et type d'annotations et d'interactions nécessaires,
- le nombre de corrections de l'annotation automatique.

De manière subjective, l'expérimentateur étant présent dans la salle d'expérimentation, il peut prendre en note certaines réactions de l'utilisateur face aux retours du système. Enfin, l'utilisateur finalise l'expérimentation en remplissant un questionnaire.

²¹ Un formulaire de consentement est signé par les utilisateurs. Il les informe qu'ils peuvent être "induits en erreur". Les utilisateurs sont finalement informés de la duperie utilisée.

C.2 Résultats

Les expérimentations ont été menées sur 7 sujets parmi lesquels des médecins. Les autres profils de sujets sont des chercheurs dans le domaine du traitement du signal ou bien de la physiologie. Tous les sujets expérimentés ont une bonne connaissance des signaux physiologiques. Pour des raisons techniques certaines traces ne sont pas complètes.

C.2.1 Évaluations quantitatives

Nous donnons les évaluations quantitatives des interprétations menées par les sujets. Le tableau 4.9 reporte les traces des activités des sujets pendant l'interprétation et le tableau 4.8 reporte une évaluation du résultat de l'interprétation à la fin de 5 minutes en le comparant avec le résultat attendu de l'annotation (*i.e.* celui qui a servi à générer les données).

		id1	id2	id3	id4	id5	id6	id7	Moyenne
Info.	s1	0,35	0,26	0,52	0,3	0,14	0,14	0,34	0,29
	s2	0,38	0,1	0,53	0,4	0,09	0,21	0,45	0,31
	s3	0,26	0	0,52	0,38	0,3	0,13	0,35	0,28
	Moy	0,33	0,12	0,52	0,36	0,18	0,16	0,38	0,29
Cont.	s1		0,1	0,15	0,37	0,18	0,09	0,26	0,19
	s2		0,13	0,19	0,3	0,1	0,15	0,17	0,17
	s3		0,14	0,1	0,27	0,12	0,16	0,23	0,17
	Moy		0,13	0,15	0,32	0,13	0,13	0,22	0,18
	<i>Prog/Info</i>		<i>0,04</i>	<i>-0,72</i>	<i>-0,12</i>	<i>-0,24</i>	<i>-0,18</i>	<i>-0,42</i>	<i>-0,27</i>
Auto.	s1	0,14	0,1	0,2	0,26	0,31	0,31	0,24	0,22
	s2	0,26	0,31	0,29	0,4	0,25	0,25	0,2	0,28
	s3	0,28	0,2	0,37	0,32	0,18	0,18	0,22	0,25
	Moy	0,22	0,2	0,29	0,33	0,25	0,25	0,22	0,25
	<i>Prog/Info</i>	<i>-0,32</i>	<i>0,69</i>	<i>-0,45</i>	<i>-0,1</i>	<i>0,42</i>	<i>0,54</i>	<i>-0,42</i>	<i>0,05</i>
	<i>Prog/Com</i>		<i>0,61</i>	<i>0,96</i>	<i>0,03</i>	<i>0,86</i>	<i>0,87</i>	0	<i>0,55</i>

Tab. 4.8.

Mesure de similarité entre le résultat de l'interprétation menée par les utilisateurs et l'annotation attendue (mesure sur la base des séries symboliques uniquement). s1, s2 et s3 correspondent aux trois séries du jeu de données interprété par un sujet avec un mode. Les *progressions* (*e.g.* *Prog/Info* = Progression par rapport au mode "informatisé") donne la progression relative de la similarité moyenne obtenue avec un mode avec un autre.

Dans le tableau 4.8, on peut constater que l'interprétation réalisée avec le mode informatisé (sans aide automatique) n'a une similarité moyenne que de ≈ 0.29 . Ceci s'explique principalement par le faible temps qui est laissé aux sujets et la non maîtrise de l'outil (induisant des pertes de temps). La difficulté de la tâche qui était demandée aurait nécessité plus de temps aux sujets. De plus, l'identification de traces d'événements a posé des difficultés et un unique sujet a pu identifier les scénarios corrects dans les enregistrements.

Les résultats obtenus avec le mode contrôlé (utilisant les algorithmes de notre système) sont globalement très inférieurs aux résultats obtenus sans automatisation (-27%). Seul le sujet id2 augmente la qualité de son annotation, mais très faiblement et à partir d'un niveau initial très faible. Ce cas ne nous semble donc pas significatif. Par rapport à nos propres expérimentations, nous expliquons ces très faibles résultats :

- par la difficulté des tâches demandées par les utilisateurs au système, Pour le premier point, nous sommes rendus compte que les sujets identifiaient souvent des regroupements de formes "élémentaires". Les symboles à apprendre étaient au delà de la complexité que pouvait supporter les modèles utilisés par le système et par conséquent, l'annotation était globalement très faible.
- par le temps passé à essayer de comprendre les annotations du système. Le résultat des annotations du système n'étaient pas toujours parfaites, les sujets ont souvent été perturbés par le résultat des automatismes. On peut illustrer ces réactions avec quelques propos recueillis sur le vif : "Je comprends pas ce qu'il fait" (id3), "Alors là il automatise n'importe quoi" (id1), "Ah non, je m'attendais

pas à ça... Qu'est-ce qu'il m'a fait ?" (id6). Par conséquent, les sujets ont systématiquement passés beaucoup de temps à essayer de comprendre ce que le système avait fait.

- par un manque de pratique de l'outil. Le manque de pratique est pointé par les utilisateurs comme une excuse des mauvaises performances globales. Après l'utilisation, la grande majorité des sujets indiquent d'eux-mêmes qu'un apprentissage préalable du système permettrait d'améliorer leur efficacité à utiliser l'interface et également à anticiper les réactions du système pour faire des annotations en ayant une connaissance de leurs implications. Pendant l'annotation, certains sujets assument une part de l'erreur d'un "mauvais" comportement de l'annotation automatique.

Du point de vue de la collaboration, les automatismes du mode contrôlé conduisent donc à l'augmentation de la charge cognitive liée à la compréhension du fonctionnement du système et donc à un rapport temps d'annotation sur temps de réflexion très faible.

Avec le mode automatique (en ligne), les résultats de la similarité sont en augmentation et regagne le niveau moyen des annotations du mode informatisé. Ces résultats ne permettent pas de conclure sur le système puisqu'il s'agit de l'expérimentateur qui interagissait avec le sujet, mais l'uniformité des résultats obtenus permet de rendre compte de l'uniformité de l'aide apportée entre les sujets. On peut donc penser qu'il n'y a pas de biais introduit par des aides plus importantes pour un sujet que pour un autre.

Le tableau 4.9 permet de juger de la différence d'activité dans les différents modes. On peut d'abord constater que les automatisations permettent de réduire en moyenne le nombre d'actions d'annoter (Total) par le sujet tout en préservant le nombre d'éléments d'annotations (Nb Annot). Pour les modes manuels et informatisés, en 5 minutes, les sujets réalisent en moyenne respectivement 30.86 et 32.5 annotations, tandis que dans les modes automatisés (contrôlé et en ligne), ils n'en réalisent respectivement que 20.5 et 21. L'automatisation permet donc de réduire d'environ 30%. Quant au nombre d'annotations sur l'interprétation finale (Nb Annot), il est en moyenne de 40.67 pour le mode contrôlé (l'augmentation significative est liée à l'intense production de symboles par le système) et de 31.33 pour le mode en ligne. L'automatisation permet donc bien de suppléer l'utilisateur dans son annotation.

Cependant, pour certains sujets en opposition avec le système, la majorité des actions menées ont été des suppressions des annotations faites par le système. Dans ce cas, les performances du système induisent une perte de temps pour le sujet. Globalement, le retour d'expérience montre que les sujets sont prêts à utiliser plus l'automatisation si le système fait moins de fautes, quitte à ce qu'il omette certaines annotations.

Dans les observations générales, on peut indiquer que la très faible utilisation du zoom est vraisemblablement liée au manque de temps des sujets pour explorer plus en finesse les signaux. Dans le mode contrôlé, différentes utilisations sont faites de l'automatisation. Certains sujets (id3, id6) utilisent beaucoup l'automatisation en réalisant très peu d'annotations, tandis que d'autres utilisateurs réalisent une grande partie de l'annotation initialement et utilisent ensuite l'automatisation. Ces derniers indiquent qu'après avoir utilisé une première fois le système il ferait, à l'avenir, plus appel à l'automatisation.

Nous avons indiqué les nombres d'annotations réalisées par série (Séries s1, s2 et s3) pour mettre en évidence l'inégalité de traitement que peuvent subir des séries temporelles à annoter dans les processus manuels. Par exemple, dans le mode informatisé, le sujet id5 n'a quasiment annoté qu'une série se privant peut être d'informations importantes dans les autres séries. Dans les modes automatiques, les annotations étant réalisées sur toute les séries par le système on cherchait à observer une réduction de ces différences de traitement. En pratique, nous les avons peu observées. La raison est certainement que le traitement de trois séries temporelles permettaient aux utilisateurs de les gérer en parallèle. Cependant, si le nombre de séries avait été beaucoup plus important comme c'est le cas envisagé pour le système, nous pensons que l'automatisation aurait montré un traitement plus équilibré de l'ensemble des séries.

C.2.2 Réponses au questionnaire

On donne ici les réponses au questionnaire rempli par les sujets à la fin de l'expérimentation.

			id1	id2	id3	id4	id5	id6	id7	Moyenne
Manuel	<i>Ajout</i>	Seg	0	0	0	0	0	0	0	0
		Symb	26	28	25	33	49	14	20	27,86
		Scen	0	0	3	3	3	6	6	3
	<i>Suppr</i>	Seg	0	0	0	0	0	0	0	0
		Symb	0	0	0	0	0	0	0	0
		Scen	0	0	0	0	0	0	0	0
	<i>Séries</i>	1	9	12	8	14	20	6	5	10,57
		2	8	10	8	12	20	7	9	10,57
		3	9	6	12	10	12	7	12	9,71
	<i>Général</i>	Total	26	28	28	36	52	20	26	30,86
		Zoom	x	x	x	x	x	x	x	x
		MAS	x	x	x	x	x	x	x	x
Info.	<i>Ajout</i>	Seg	8				0	0	6	3,5
		Symb	33				24	15	28	25
		Scen	6				2	0	3	2,75
	<i>Suppr</i>	Seg	0				0	0	0	0
		Symb	2				2	1	0	1,25
		Scen	0				0	0	0	0
	<i>Séries</i>	1	21				4	3	14	10,5
		2	14				3	10	12	9,75
		3	14				21	3	11	12,25
	<i>Général</i>	Total	49				28	16	37	32,5
		Zoom	0				0	0	0	0
		MAS	x	x	x	x	x	x	x	x
Cont.	<i>Ajout</i>	Seg	8		5	16		7	9	9
		Symb	2		0	0		3	3	1,6
		Scen	0		0	0		0	0	0
	<i>Suppr</i>	Seg	24		0	0		7	4	7
		Symb	2		0	0		5	4	2,2
		Scen	0		0	0		0	0	0
	<i>Séries</i>	1	18		6	6		9	12	10,2
		2	15		2	6		9	4	7,2
		3	7		0	4		4	4	3,8
	<i>Général</i>	Total	38		6,5	16		22	20	20,5
		Zoom	0		0	0		0	0	0
		MAS	10		24	8		14	8	12,8
Nb Annot			45	16	72	78	19	14	40,67	
Auto.	<i>Ajout</i>	Seg	0	0	0	1	17		0	3
		Symb	8	13	13	13	15		4	11
		Scen	4	0	0	0	0		0	0,67
	<i>Suppr</i>	Seg	0	0	0	3	1		0	0,67
		Symb	5	0	6	21	0		0	5,33
		Scen	0	0	0	0	0		0	0
	<i>Séries</i>	1	7	4	0	11	9		2	5,5
		2	6	6	6	17	8		2	7,5
		3	6	3	15	10	16		0	8,33
	<i>Général</i>	Total	18	13	20	38	33		4	21
		Zoom	0	0	2	0	0		0	0,33
		MAS	x	x	x	x	x	x	x	x
Nb Annot		19	44	23	33	37		32	31,33	

Tab. 4.9.

Évaluations quantitatives de l'activité des utilisateurs devant le système. Pour chaque mode, on donne les annotations réalisées (ajouts et suppressions) par type d'annotation et par série. L'encart 'Général' donne la somme des annotations réalisées par le clinicien mais également identifie l'utilisation du Zoom, compte les appels effectués au système (MAS) et le nombre d'annotations finales obtenues (humains et systèmes confondus). Les absences de valeurs sont notifiées par des absences de valeurs. Les 'x' indiquent qu'aucune valeur n'est attendue.

C. Évaluation de la pertinence de la collaboration

	Manuel	Informatisé	Contrôlé	Automatisé
Quel mode vous semble le plus adapté pour faciliter l'interprétation des séries temporelles ?	0	6	0	1
Quel mode vous semble le plus adapté pour faciliter l'annotation des séries temporelles ?	1	1	2	3

On donne dans ce tableau le nombre de réponses données par les sujets par mode. On voit au travers de la première question que l'assistance informatique dans une tâche d'interprétation, *i.e.* l'aide à la compréhension des séries temporelles, n'est pas une chose aisée. Dans les évaluations courtes qui sont menées, les utilisateurs n'ont pas le temps de s'adapter au système. Il passe donc beaucoup de concentration à essayer de comprendre comment mener à bien les annotations qu'ils désirent faire. On observe que dans les deux modes automatisés, les utilisateurs perdent le fil de la tâche qu'ils ont à faire et oublient l'annotation des scènes. Malgré le fonctionnement aidé par un humain en Magicien d'Oz, seul 1 utilisateur (id3) a attribué la capacité à faciliter l'interprétation au dernier mode automatique (On notera dans le tableau 4.8 qu'il s'agit du sujet qui a le plus bénéficié du mode automatique).

Par contre, pour une tâche plus systématique telle que l'annotation, les automatisations sont plus acceptées. Bien que le système fasse des erreurs dans l'automatisation (mode contrôlé), ils laissent entrevoir aux utilisateurs des capacités d'automatisation de la tâche d'annotation. Le mode automatique est alors préféré au mode contrôlé car réalisant moins de fautes.

Perception des utilisateurs sur le système dans les différents modes Plusieurs réponses possibles, on indique dans le tableau le cumul des attributions. Initialement, il existait deux catégories pour assistant (assistant et compagnon) et deux catégories pour Collaborateur (Collaborateur et partenaire). Nous les avons regroupées car les sujets ne les différenciaient pas.

	Outil	Assistant	Collaborateur	Compétiteur
Informatisé	4	1	0	0
Contrôlé	0	3	4	2
Automatique	0	1	4	2

De manière générale, on retrouve dans cette question une diagonale attendue. Le dernier fonctionnement (compétiteur) n'avait pas été envisagé lors de la conception du système. Dans les remarques recueillies, on note l'existence de deux types de compétition :

- l'émulation entre l'utilisateur et le système qui invite l'utilisateur à essayer de faire mieux que le système.
- une compétition dans laquelle le système fait une autre interprétation, contrariante pour l'utilisateur.

Le sujet id6 indique "Quand je veux mettre du 2 et qu'il me met du 3, alors là ça m'énerve!".

Si le premier mode a des effets positifs sur le résultat de la collaboration en poussant à l'utilisateur à explorer les données, le second est réhibitoire pour l'utilisation du système.

Évaluation des éléments de l'annotations

	Oui (%)	Non (%)
Les éléments d'annotations permettent-ils de construire une interprétation des séries temporelles ?	93	7

La question cherche à savoir si les annotations sous la forme de segments, symboles et scènes permettent de construire une interprétation, *i.e.* permettent d'expliquer ce qui s'est passé sur les séries temporelles. Ces concepts ne sont pas usuellement utilisés et donc les utilisateurs en ont finalement peu de recul. Il est intéressant cependant de remarquer que l'un des utilisateurs a été amené à se poser la question de la signification de ces concepts pendant son interprétation. Il s'est en effet intéressé à comparer les bruits dans les enregistrements (bien qu'il s'agisse d'un bruit gaussien uniforme). On voit donc ici la difficulté de la manipulation de ces concepts d'annotation dans leur forme brute. L'utilisateur qui annote selon le bruit alors que le système fonctionne sur les formes ne peut espérer obtenir satisfaction par les automatismes.

Sur la notion de scénario, certains sujets ont été surpris de voir qu'un scénario pouvait impliquer deux évènements sans inclure un évènement qui s'est passé entre les deux.

Évaluation du mode commandé

	Oui (%)	Non (%)
Les actions menées par le système sont elles prévisibles ?	64	26
Souhaiteriez vous qu'elles soient (plus) prévisibles ?	100	0
Avez vous réussi à réaliser l'annotation souhaitée des séries temporelles ?	42	58
Estimez vous l'annotation plus efficace avec l'automatisation des traitements ?	43	57

Les automatismes du système sont jugées prévisibles par les utilisateurs mais dans tous les cas il semble nécessaire de faire en sorte qu'il le soit plus. Il est à noter ici que dans le mode commandé, les utilisateurs utilisaient des traitements des agents non bouclés, c'est-à-dire que les agents ne réalisaient qu'un seul pas de leur traitement et recommençaient de zéros la fois suivante. Il n'y avait donc pas de bouclage comme l'expérimentation précédente.

Globalement les utilisateurs voient dans le système un "collaborateur maladroit" (id1) ou "encombrant" (id4). Et la maladresse du système perturbe les sujets pendant leur tâche en leur faisant parfois perdre le fil de leur raisonnement propre, mais elle est parfois acceptée dans la mesure où le système peut amener quelque chose de plus à l'utilisateur : "le système a fait des erreurs, par omission, mais il m'a également montré que j'avais fait des erreurs" (id7). Le système serait mieux accepté d'une part si les annotations du système étaient mieux distinguées de celles de l'utilisateur et si les annotations étaient accompagnées d'une mesure de confiance donnée par le système. Il est plus acceptable pour un utilisateur de voir une erreur si celle-ci est indiquée avec une mesure de confiance faible du système.

Cependant, dans l'état actuel, les automatismes sont un frein au bon déroulement de l'annotation pour l'utilisateur qui n'arrive pas toujours à réaliser l'annotation souhaitée et qui par conséquent trouve le mode informatisé plus efficace.

Évaluation du mode en ligne

	Oui (%)	Non (%)
Les interventions du système sont elles pertinentes ?	71	29
Les interventions du système sont elles faites au bon moment ?	60	40
Diriez vous que vous avez subit les actions du système ?	57	33
Acceptez vous le fait que le système puisse proposer une interprétation différente de la votre ?	86	14
Avez vous rencontré des difficultés à comprendre les intentions du système ?	33	57

Dans le cadre d'une expérimentation en Magicien d'Oz, les questions cherchent à évaluer le mode d'interaction envisagé pour notre couplage structurel, *i.e.* par l'échange des annotations. L'expérimentateur participe à aider l'annotation de l'utilisateur, mais il introduit parfois volontairement des erreurs et réalise l'annotation de nouvelles formes simples telles que les aplats. Cela permet de s'approcher du fonctionnement réel du système.

Les réponses aux deux premières questions vont tout à fait dans le sens de l'acceptation d'un mode de communication par des annotations réalisées en parallèle de l'interprétation. Cependant, les utilisateurs disent subir le système. Dans l'état actuel, le système s'obstine parfois dans une voie de laquelle il ne peut sortir, c'est ce qui a amené certains utilisateurs à juger le système comme un compétiteur. Ce comportement était également reproduit dans l'expérimentation du mode en ligne. Les utilisateurs qui disent subir le système pointent le fait qu'ils souhaitent garder la maîtrise de l'annotation réalisée. C'est-à-dire que les modifications qui sont réalisées doivent être prises en compte. Donc, il semble qu'au moyen des seuls annotations, il n'y a pas de négociation possible. Le cas extrême étant un utilisateur qui n'accepte pas que le système puisse proposer une interprétation différente dans la mesure où c'est lui l'"expert".

En grande majorité, le fait que le système puisse proposer une interprétation différente est accepté par les utilisateurs mais à certaines conditions :

- l'affichage d'une incertitude du système,
- la possibilité d'imposer son choix,
- la possibilité de valider ou non ces interprétations,

- la “pertinence” de l'interprétation,
- la “crédibilité” de l'interprétation.

La question de compréhension des intentions du système cherchait à mettre en évidence la difficulté du système à faire comprendre sa “cohérence interne” qui justifie ces annotations. Cette difficulté de compréhension transparaît dans certaines réactions ponctuelles des utilisateurs. On s'attendait donc à ce que les utilisateurs indiquent leur incompréhension des intentions du système. Mais dans le questionnaire, les utilisateurs ont estimé en moyenne qu'ils comprenaient les “intentions” du système²².

Avantages et inconvénients des différents modes : Finalement, on dispose d'une liste des avantages et des inconvénients des différents modes qui permettent de mettre en avant leurs spécificités.

- Mode manuel :
 - Avantages : lisibilité, maîtrise totale, rapide, précision
 - Inconvénients : stockage, peu pratique pour corriger, pas de trace informatique
- Mode info :
 - Avantages : intuitif, zoom, accès direct au signal intéressant, corrections faciles, précision
 - Inconvénients : incertitudes, familiarisation avec logiciel, parfois difficile de “voir”
- Mode contrôlé :
 - Avantages : reconnaissance d'évènements tronqués, gain de temps sur certains évènements
 - Inconvénients : exactitude de la reconnaissance, perte de temps si erreurs, fournis trop d'information en même temps, ne tient pas compte des corrections, difficulté de reproductibilité, difficulté avec les scénarios
- Mode automatique en ligne :
 - Avantages : pour les signaux longs, gain de temps sur certains évènements, très réactif
 - Inconvénients : erreurs, impose de chercher à comprendre la démarche du logiciel

D Utilisation pour l'interprétation de séries temporelles de patients ventilés

Nous avons finalement utilisé notre système pour explorer les données de patients ventilés afin d'extraire des modèles d'évènements et de scénarios intéressants. Le système étant un prototype, l'expérimentation n'a pas été menée par des cliniciens seuls capables de juger de l'intérêt pratique des évènements et modèles construits par le système. Nous avons déjà pu montrer dans les évaluations préliminaires que les modèles d'évènements et de scénarios ont un caractère générique inter-patients. Nous avons donc pu explorer les séries temporelles multivariées indépendamment pour chaque patient. Ne traiter qu'une série à la fois nous a permis d'avoir des temps de calcul raisonnables pour mener une interprétation collaborative dynamique avec le système.

Nous relatons ici l'expérimentation menée sur les séries temporelles d'un seul patient. Cet exemple est représentatif des comportements qui sont observés pour l'exploration des différentes séries temporelles des patients à disposition.

Dans un premier temps, nous avons réalisé une annotation à partir de quelques cycles entiers d'un patient. Les figures 4.29 et 4.32 montrent les résultats obtenus pour une partie du signal et les figures 4.30 et 4.31 donnent les modèles qui ont été appris (modèles de la forme d'un segment).

On peut voir que l'annotation du signal de pression est plus facilement réalisée que celui du signal de débit. Dans les deux cas, le système a bien identifié deux types de symboles à partir des quelques annotations réalisées (4 cycles ont été annotés). Les symboles 1 (pour la pression comme pour le débit) sont associés à des cycles pendant lesquels il y a eu un effort inefficace. Conformément aux descriptions des efforts inefficaces [159], on constate sur les modèles des figures 4.30 et 4.31 que le débit le cycle présente un plateau et sur la pression, il y a une surpression en fin d'expiration.

Par conséquent, les modèles qui ont été construits par le système pourraient être utilisés pour la détection en ligne des futures asynchronies que le patient pourrait subir.

²²On peut faire également la remarque qu'il n'y a eu aucune gêne pour les sujets à parler de l'“intention” d'une machine.

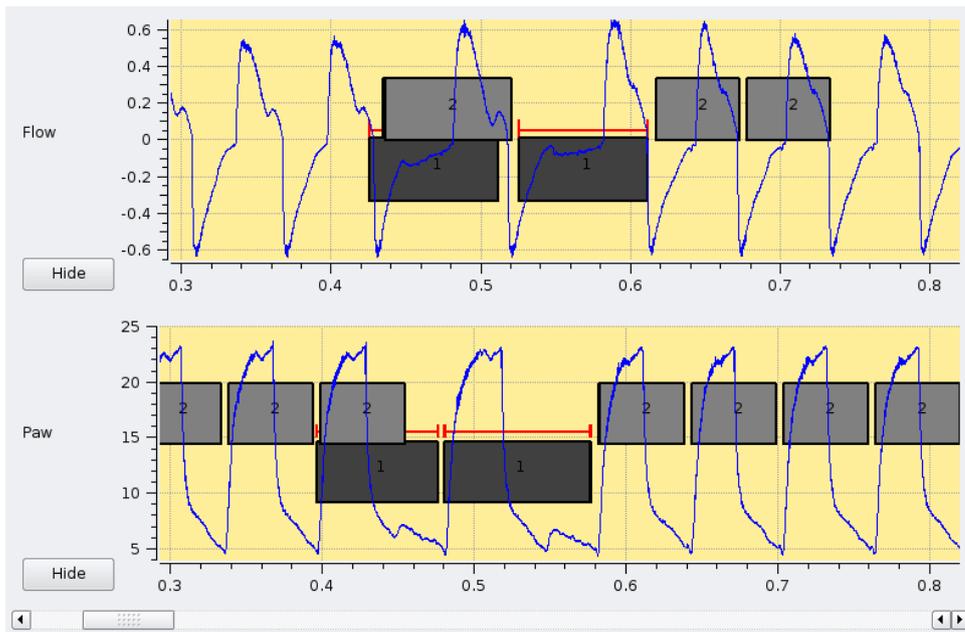


FIG. 4.29. Annotations des débits et pressions respiratoires d'un patient présentant des asynchronies P/V.

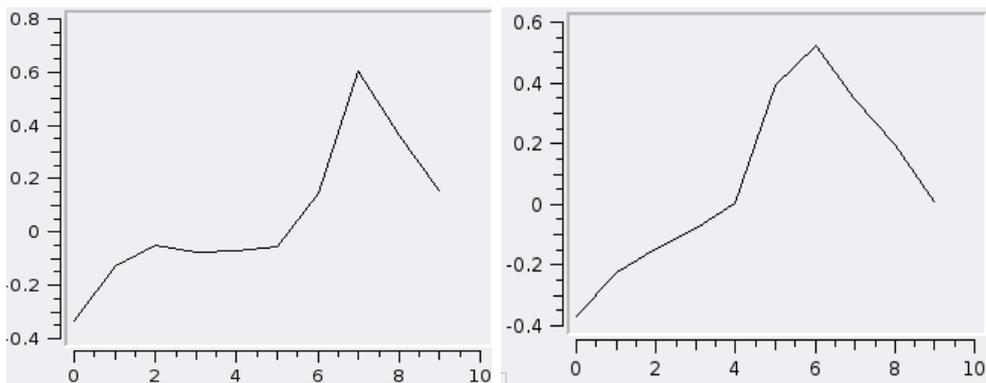


FIG. 4.30. Modèles des cycles du débit respiratoire. Sur la gauche, le modèle 1 correspond à un modèle de cycles asynchrones et sur la droite le modèle 2 à un cycle normal. En abscisse, on lit les positions successives des 10 points du modèle de forme.

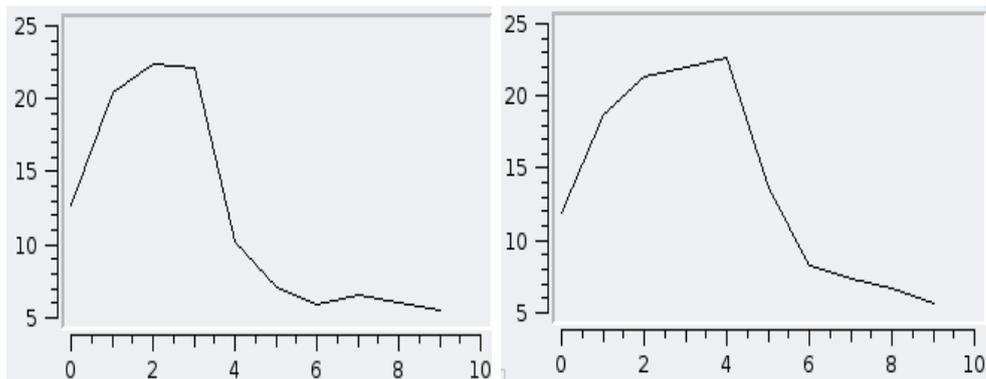


FIG. 4.31. Modèles des cycles de la pression respiratoire. Sur la gauche, le modèle 1 correspond à un modèle de cycles asynchrones et sur la droite le modèle 2 à un cycle normal. En abscisse, on lit les positions successives des 10 points du modèle de forme.

D. Utilisation pour l'interprétation de séries temporelles de patients ventilés

L'apprentissage de scénarios ²³ à partir de cette symbolisation des séries temporelles donne les résultats suivants :

Explication de (0 2) : (1,1, [-0.22,0.06]), (0,2, [-0.27,0.01]), (0,2, [-0.27,0.01])

Explication de (0 1) : (1,1, [-0.25,0.06]), (0,2, [-0.30,-0.08]), (0,1, [-0.29,0.08]), (0,1, [-0.29,0.08])

Explication de (1 2) : (0,2, [-0.28,0.05]), (0,2, [-0.24,0.05]), (0,1, [-0.29,0.03]), (0,1, [-0.25,0.04])

Explication de (1 1) : (1,2, [-0.30,-0.06]), (1,2, [-0.30,0.00])

La lecture d'un scénario est faite comme précédemment : il s'agit d'une suite d'évènements avec leurs contraintes temporelles. Mais ici, un évènement est défini par deux identifiants : le type de la série, 0 pour le débit ou 1 pour la pression, et l'identifiant du symbole. Ainsi les évènements sont les suivants :

- (0 1) est l'identifiant de l'évènement 1 sur le signal de débit (Cycle de débit d'un effort inefficace).
- (0 2) est l'identifiant de l'évènement 2 sur le signal de débit (Cycle de débit normal).
- (1 1) est l'identifiant de l'évènement 1 sur le signal de pression (Cycle de pression d'un effort inefficace).
- (1 2) est l'identifiant de l'évènement 2 sur le signal de pression (Cycle de pression normal).

Les scénarios traduisent donc qu'avant un cycle d'effort inefficace annoté sur le débit (0 1), il y a dans 70% des cas deux mêmes évènements qui le précèdent, un cycle normal ainsi qu'une asynchronie annotée sur le signal de pression. Les contraintes temporelles qui sont apprises sont finalement peu informatives puisque les répétitions des évènements conduisent à des contraintes qui sont les contraintes fixées par la taille de la fenêtre d'apprentissage. Un cycle d'asynchronie identifié sur le signal de pression est précédé de deux cycles normaux. Cette différence dans le scénario provient de la meilleure sensibilité de la reconnaissance de cycles normaux pour le signal de pression. Beaucoup de cycles normaux n'ont pas été annotés sur le signal de débit.

La figure 4.32, montre le résultat de la reconnaissance des scénarios pour l'annotation des scènes.

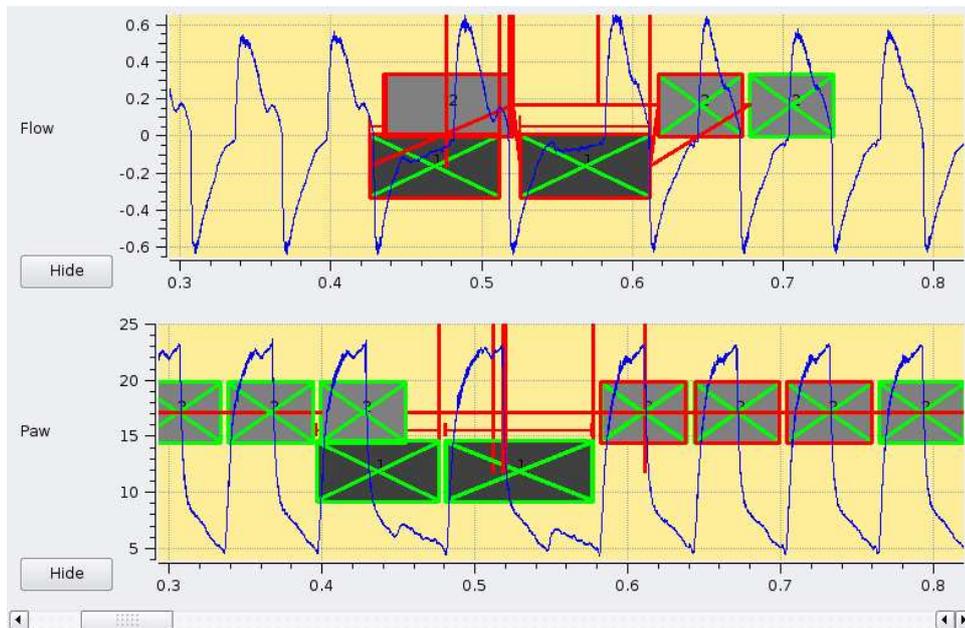


FIG. 4.32.

Annotations des débits et pressions respiratoires d'un patient présentant des asynchronies P/V avec les scènes. Les barres rouges verticales indiquent la suite des liens entre les symboles de deux séries différentes.

Nous avons ensuite fait de nouveau appel au système de sorte à ce qu'il affine ses annotations et ses modèles. Nous avons pu alors constater l'apparition d'un nouveau type de cycle sur le signal de débit. Les annotations du signal de pression sont très stables. La figure 4.33 illustre le résultat de l'annotation du système après quelques cycles successifs, les modèles pour les évènements du signal de débit sont donnés dans la figure 4.34

²³L'apprentissage a été réalisé avec une taille de fenêtre de 0.3, soit environ trois cycles pour ce patient, et un seuil de fréquence de 70%.

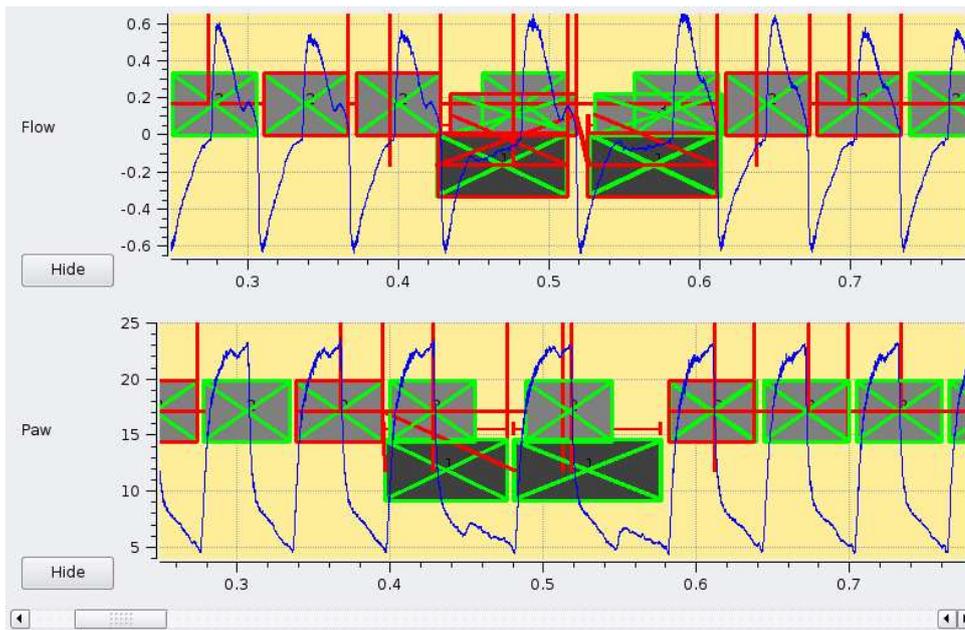


FIG. 4.33. Annotations des débits et pressions respiratoires d'un patient présentant des asynchronies P/V avec les scènes. Les barres rouges verticales indiquent les liens entre les symboles de deux séries différentes.

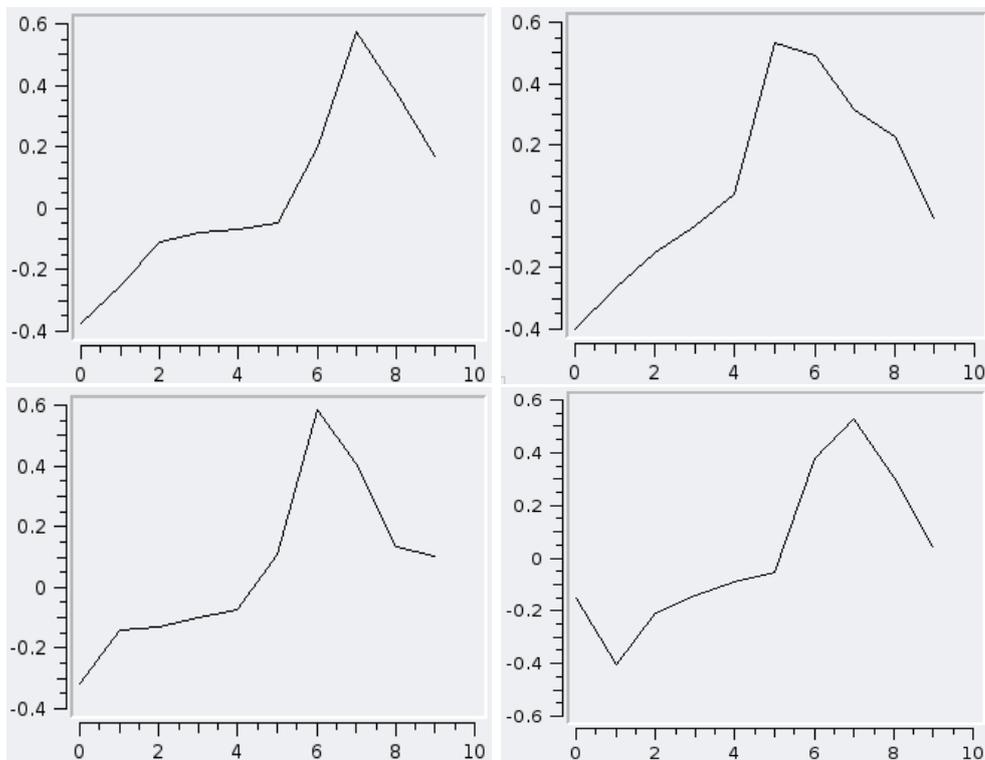


FIG. 4.34. Modèles des cycles de la pression respiratoire après plusieurs exécutions du système. En haut à gauche, le modèle 1 correspond à un modèle de cycles asynchrones, en haut à droite le modèle 2 correspond à un cycle normal. En bas, deux nouvelles formes de cycles asynchrones, à gauche, le modèle 3 est étendu sur la droite par une pente réduite, à droite, le modèle 4 est étendu sur la gauche par une fin de cycle précédent. En abscisse, on lit les positions successives des 10 points du modèle de forme.

D. Utilisation pour l'interprétation de séries temporelles de patients ventilés

Les scénarios qui ont été appris sont les suivants :

Explication de (0 4) : (1,2, [-0.30,0.03]), (1,1, [-0.23,0.07]), (0,1, [-0.28,0.00]), (0,3, [-0.28,0.01]), (0,3, [-0.27,0.01]), (0,1, [-0.29,0.00])
Explication de (0 3) : (1,2, [-0.28,0.04]), (1,2, [-0.29,0.04]), (1,2, [-0.26,0.03]), (0,2, [-0.30,-0.06]), (0,2, [-0.30,-0.06])
Explication de (0 2) : (1,2, [-0.30,0.03]), (1,2, [-0.29,0.03]), (1,1, [-0.30,0.06]), (0,1, [-0.29,0.08]), (0,1, [-0.29,-0.00])
Explication de (0 1) : (1,2, [-0.29,0.03]), (1,2, [-0.29,0.03]), (1,1, [-0.29,0.06]), (0,4, [-0.25,0.08]), (0,2, [-0.28,-0.00])
Explication de (1 2) : (1,2, [-0.27,-0.02]), (1,2, [-0.26,-0.03]), (0,1, [-0.29,0.04]), (0,3, [-0.29,0.04]), (0,2, [-0.26,0.04]), (0,1, [-0.29,0.04])
Explication de (1 1) : (1,2, [-0.30,0.00]), (0,2, [-0.30,0.05]), (0,2, [-0.29,0.05])

On constate tout d'abord sur la figure 4.33 que le système a amélioré son annotation des cycles normaux sur le signal de débit puisqu'ils sont tous annotés par le symbole 2 sur la partie du signal de l'illustration. Ensuite, on constate que les modèles 1 et 3 servent à annoter les mêmes cycles. Par conséquent, le système fait une différenciation non compréhensible à partir de cette illustration. Il faudrait retrouver dans les séries les annotations exclusives de l'un ou l'autre des événements pour savoir si il s'agit d'une différenciation intéressante ou bien s'il faut demander au système de ne faire qu'un seul type d'évènements (au moyen de liens entre les symboles).

Les scénarios et les scènes ne sont pas beaucoup plus intéressants que dans le cas précédent. Toujours avec la même configuration, les contraintes temporelles n'apportent pas beaucoup d'information et les scénarios ne nous semblent pas pointer de comportement intéressant.

Dans un second temps, nous avons réalisé une annotation alternative des séries temporelles. À la place d'annoter des cycles entiers, nous avons uniquement indiqué au système des parties de cycles qui nous semblaient intéressantes et dont nous cherchions à savoir si elles pouvaient être caractéristiques des cycles d'asynchronies. La figure 4.35 illustre les différents types d'annotations qui ont été réalisées par le système (à partir de quelques segments). Pour le signal de pression, nous nous sommes contentés d'annoter les surpressions de fin d'expiration, et pour le signal de débit, nous avons annoté les plateaux des cycles asynchrones et des augmentations du débit qui peuvent être parfois observés.

À la suite de ce premier retour du système, nous avons aidé le système à détecter plus facilement certains plateaux en ajoutant des segments lorsqu'ils étaient manquants. Une fois l'annotation souhaitée réalisée, on exécute l'apprentissage de scénarios (*cf.* figure 4.36 pour les scènes identifiées). Les scènes visualisées sur cette figure sont plus claires que les scènes des figures précédentes. On y distingue en particulier bien l'explication de la "bosse" du débit qui est précédée d'un plateau sur le débit et de trois surpressions ((1,1, [-0.29,-0.01]), (1,1, [-0.29,0.01]), (1,1, [-0.29,0.01]), (0,1, [-0.28,0.00])). On voit en particulier que les contraintes temporelles trop larges conduisent à annoter des scènes sur les troisième, quatrième et cinquième symboles (0 2) avec dans les mêmes symboles (0 1) et (1 1) qui interviennent dans la scène.

Pour la question que nous nous sommes initialement posée comme objectif de l'exploration, nous voyons ainsi par l'annotation réalisée que le symbole (0 2) n'est pas caractéristique d'une asynchronie puisque la scène faisant contenir un plateau et une "bosse" sur le signal de débit n'est pas systématiquement identifiée. Un tel résultat inviterait un clinicien à s'interroger sur la présence d'une telle "bosse" et en particulier savoir si elle peut également être identifiée sur d'autres patients.

Les expérimentations qui ont été menées à partir des données de patients ventilés montrent donc la capacité du système d'une part à construire des modèles d'évènements qui pourraient être utilisés pour la détection en ligne des asynchronies et d'autre part à faciliter l'annotation et le questionnement d'un utilisateur pour le pousser à l'identification de caractéristiques qu'il n'aurait peut-être pas initialement envisagées comme significatives.

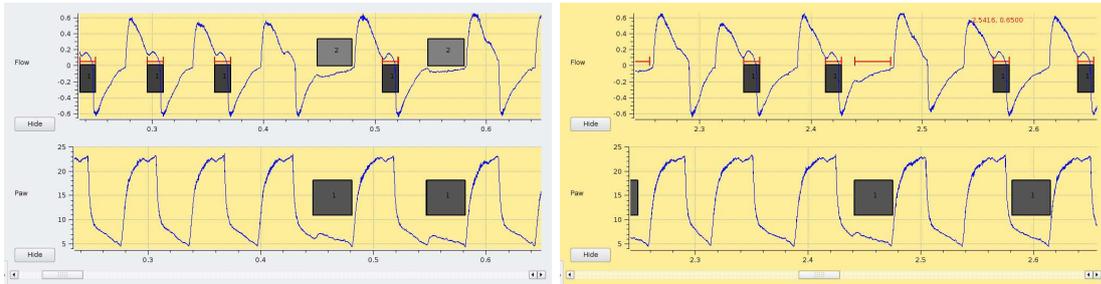


FIG. 4.35. Annotation alternative des débits et pressions respiratoires. À droite, on donne l'annotation d'origine, qui est ensuite complétée par l'introduction de segments (à droite) pour faciliter le traitement du système.

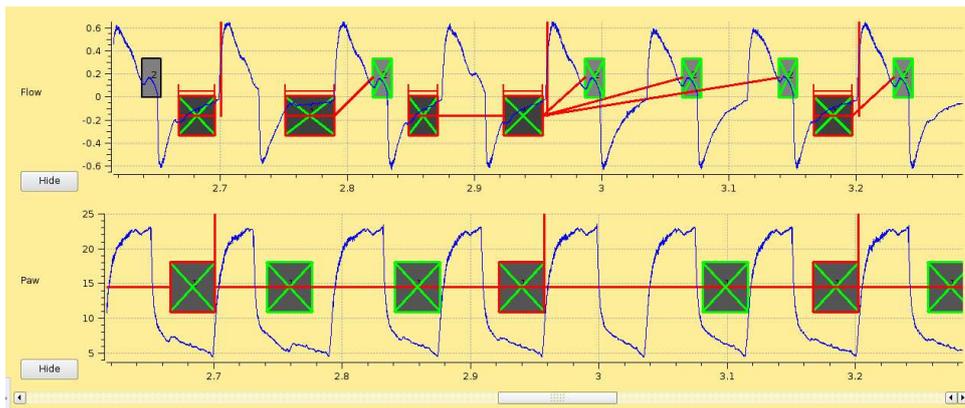


FIG. 4.36. Annotation alternative des débits et pressions respiratoires complétée par de nouveaux segments.

Chapitre 5

Discussions et perspectives

À la lueur des évaluations qui ont été menées, nous discutons l'utilisation du paradigme de l'Enaction et des annotations pour la proposition d'un système permettant de réaliser collaborativement l'interprétation de séries temporelles. Nous pointons en particulier des difficultés liées au manque d'intention du système et au principe de la construction d'un consensus entre les interpréteurs. Cependant, il semble intéressant de poursuivre l'étude de l'approche Enactive pour la conception de systèmes collaboratifs.

Dans un second temps, nous revenons sur les solutions techniques qui ont été proposées pour réaliser notre système. Ces techniques sont calculatoires mais répondent aux problèmes complexes qu'elles adressent en offrant une expressivité importante à l'interprétation réalisée. Deux dimensions pourraient être ajoutées pour compléter l'expressivité de l'interprétation, la prise en compte de la granularité temporelle et la prise en compte de la dimension contextuelle des données.

Finalement, nous revenons sur notre problématique applicative en cherchant à identifier les domaines biomédicaux dans lesquels notre système peut amener une aide aux cliniciens. Et nous ouvrons la perspective de valorisation des méthodes de traitement automatique sous la forme d'un système d'aide à l'annotation de séries temporelles.

La travail que nous avons présenté autour de la question de l'interprétation collaborative de séries temporelles comporte trois dimensions qui peuvent être discutées à la lueur des évaluations que nous avons menées.

- La dimension conceptuelle nous a amené à s'inspirer du paradigme de l'Enaction pour concevoir la collaboration comme un couplage structurel entre deux systèmes autonomes et adaptatifs.
- La dimension opérationnelle nous a conduit jusqu'à l'implémentation d'un système et d'algorithmes pour la conception d'un prototype fondé sur les caractéristiques des systèmes Enactifs.
- La dimension applicative a été un moteur pour le choix d'une approche collaborative (dans le but d'extraire des modèles d'évènements et de scénarios dans des séries temporelles) et également un objectif qui a participé à l'opérationnalisation du système.

À partir de ces trois dimensions, on peut dresser trois conclusions globales sur notre travail :

- En s'inspirant du paradigme de l'Enaction nous avons abordé la question de la construction de modèles d'évènements et de scénarios de manière originale en proposant d'une part de faire réaliser cette tâche par l'exécution d'une interprétation de séries temporelles, et d'autre part de faire réaliser l'interprétation de manière collaborative.
- La conception et la réalisation d'un système nous permet de montrer la faisabilité d'une approche Enactive de la collaboration sur le plan technique. Mais notre conception s'est focalisée sur la réalisation d'un couplage structurel entre un système et un humain, et du point de vue de l'humain, les expérimentations ont permis de révéler des manquements à l'efficacité possible d'une telle collaboration (problème de compréhension des intentions du système, problème de réactivité du système ou problème à vouloir partager la même interprétation).
- Les premiers retours sur l'utilisation du système sur des données médicales montrent l'intérêt d'aider à l'interprétation de signaux physiologiques avec un système proposant sa propre interprétation. Mais la difficulté reste très importante et pour être plus démonstratif, les données d'application de notre système doivent être mieux choisies.

Dans la suite de ce chapitre, nous revenons plus en détails sur quelques difficultés qui ont été identifiées. Ces limites posent parfois des problèmes théoriques et pratiques importants. Elles permettent de mettre en perspective notre approche selon les trois dimensions précitées. Pour certaines de ces difficultés, nous proposons également des perspectives permettant d'ouvrir notre travail vers de nouvelles pistes.

A L'Enaction et l'annotation comme approche de l'interprétation collaborative

L'approche inspirée de l'Enaction apporte un point de vue original sur l'interprétation collaborative. Elle permet de reconnaître le besoin d'autonomie des acteurs dans une tâche d'interprétation et de pousser à conserver cette autonomie dans une réalisation collaborative de cette interprétation.

Les expérimentations montrent l'intérêt de poursuivre l'exploration de cette voie. Les utilisateurs montrent un intérêt certain à disposer d'un système capable de proposer une interprétation alternative (*e.g.* identifier de nouveaux motifs ou montrer les erreurs réalisées). Par les expérimentations en Magicien d'Oz, on a pu voir que l'approche d'une interprétation menée collaborativement et dynamiquement avait des effets incitatifs à l'approfondissement de l'exploration des données. Par conséquent, l'approche Enactive nous a permis de développer un système qui dépasse la simple aide extérieure à la réalisation d'une tâche, et elle a permis de créer une situation motivante pour l'utilisateur.

Par conséquent, l'Enaction et l'annotation donnent une approche de l'interprétation collaborative intéressante. Mais aux vues des expérimentations, il semble nécessaire de dépasser l'application assez exclusive du couplage structurel pour y intégrer des fonctionnalités réflexives qui rendrons le système plus fonctionnel. Nous pouvons illustrer ceci par le problème de la compréhension de l'intention du système et le problème de la recherche du consensus par le couplage.

A.1 Introduire de l'intention dans le système

Les expérimentations d'utilisation du système ont montré qu'un utilisateur avait des difficultés à appréhender les modifications introduites par le système d'une part parce qu'il ne les comprend pas toujours et d'autre part parce qu'il ne peut pas les anticiper. Il nous semble que les difficultés rencontrées par les

utilisateurs sont liées au manque d'intention du système. En effet, en tant que système autopoïétique, le système s'adapte en *réaction* à des perturbations de l'utilisateur et non avec une intention. En particulier, le mode d'interaction par des annotations serait intentionnel si la machine prenait en compte le fait que la modification de son interaction allait avoir ces conséquences sur l'interprétation de l'utilisateur.

Pour les utilisateurs, l'annotation comme simple marque est une information insuffisante. L'ajout d'informations plus riches (*e.g.* sur son incertitude, sa pertinence, sa conviction ou sur les raisons de son introduction ...) lui permettrait de mieux estimer l'intérêt de celle-ci. On peut ajouter une graduation de la confiance qu'a le système dans les annotations qu'il propose. Ainsi, l'utilisateur percevrait une forme d'intention de la part du système. Symétriquement, l'utilisateur peu confiant dans son annotation pourrait également le signaler. De plus, l'utilisation d'une confiance par le système permettrait de filtrer les éléments à présenter à l'utilisateur : il conserverait ainsi certains éléments uniquement pour lui en attendant qu'il en ait une confiance suffisante pour le présenter à l'utilisateur. Afin de retranscrire une forme d'intention, d'autres mécanismes permettraient d'enrichir les annotations brutes avec des informations sur la façon dont elles ont été produites.

Les systèmes collaboratifs à initiative mixte proposent également des solutions à ce problème. Un système à initiatives mixtes agit pour accomplir la tâche commune d'une part en planifiant son action prenant en compte les spécificités de l'utilisateur et d'autre part en étant capable d'expliquer à l'utilisateur ces actions lorsque celui-ci le demande. La planification et l'utilisation d'explication permettent de donner le sentiment chez l'utilisateur d'une forme d'intention du système dans ses actions.

Actuellement, le fait d'avoir des changements brusques (*transitions de phases*) ou bien des comportements cycliques peut s'avérer perturbant pour les utilisateurs. Ceci donne à l'utilisateur un sentiment d'irrationalité des traitements. En conséquent, on ne peut prêter aucune intention au système. Il ne s'agit pas de remettre en cause la démarche de conception d'un système fondé sur l'Enaction, mais d'englober un tel système d'une couche de mécanismes *réflexifs*.

En introduisant une couche de traitement supplémentaire qui observe le déroulement de l'interprétation, le système pourrait de lui-même détecter les comportements irrationnels et y remédier afin d'assurer d'une part une impression de continuité des traitements et d'autre part de pouvoir amener des explications au déroulement qui l'a conduit à réaliser une annotation. Le système pourrait par exemple disposer :

- d'un historique de ses actions de sorte :
 - qu'il puisse les annuler lui-même et réaliser ainsi des essais/erreurs en propre avant de les livrer à l'utilisateur,
 - qu'il puisse donner une explication du cheminement logique qui l'a conduit à proposer des modifications.
- de mécanismes d'analyse de ses propres activités permettant par exemple d'identifier des bouclages,
- de mécanismes d'analyse de l'activité du l'utilisateur de sorte à pouvoir anticiper ses actions et l'aider ainsi au mieux,
- de planification des actions à partir d'une situation en cours (*i.e.* le problème courant et l'activité de l'utilisateur).

Il y a derrière ces quelques mécanismes des difficultés très importantes pour arriver à une réalisation technique effective. Ces fonctionnalités que nous proposons ici de mettre en place pour combler certains déficits de notre système nécessite d'avoir des mesures permettant de qualifier les propriétés du processus d'interprétation. De telles mesures fiables sont difficiles à définir. Par exemple, lors des évaluations, la mesure des distances entre annotations successives pourraient être l'une d'elles. Elle devait permettre de mettre en évidence la stabilisation de l'interprétation. Mais en pratique, cette mesure pourrait difficilement être utilisée pour que le système décrète automatiquement la fin de l'interprétation.

A.2 La recherche d'un consensus dans l'interprétation

Un second problème rencontré dans l'approche de la collaboration proposée est celui du consensus. Nos mécanismes d'adaptation sont fondés sur la résolution des situations de contradiction entre les éléments de l'annotation (recherche d'une cohérence). Ce fondement induit le fait que la collaboration entre le système et l'humain se fasse dans l'objectif de construction d'un consensus : les interprétations de chacun doivent s'accepter l'une l'autre. Ceci ne revient pas à dire qu'elles sont les mêmes, mais qu'elles ne peuvent pas rentrer en contradiction.

Le problème du consensus est qu'il nécessite souvent des compromis dans lesquels se noient les spé-

cificités des points de vue de chaque partie. La recherche d'un consensus n'est donc pas nécessairement souhaitable pour la réalisation d'une interprétation. En effet, la construction d'un consensus est une incitation à la systématique uniformisation des points de vue alors qu'une interprétation peut nécessiter d'être en conflit prolongé avec les autres points de vues avant de faire émerger une nouvelle interprétation valide.

Il nous semble donc important de relaxer la contrainte de construction d'une interprétation cohérente entre le système et le clinicien pour tenir compte de la variabilité des interprétations :

1. entre patients : la construction de groupe de patients a été évoqué dès le début de ce travail comme contexte de l'interprétation des séries temporelles. Les groupes de patients pouvant être vus comme un nouveau niveau supérieur aux niveaux des scénarios que l'interprétation devrait construire et caractériser (*cf.* section B.1.1).
2. entre cliniciens : on ouvre ainsi une extension possible du système pour qui consisterait à se servir de l'annotation pour la confrontation de différents points de vue sur le patient. Chaque clinicien disposant de son système formé par sa propre connaissance, ces systèmes pourraient servir de liens entre les cliniciens. Un fonctionnement collaboratif des systèmes permettrait de pointer aux cliniciens, ceux de leurs collègues qui peuvent apporter des réponses à leurs questions.

A.3 Schémas Sensori-moteur : Faire des essais/erreurs

Si on veut renforcer le couplage, il faut réussir à faire construire au système et à l'utilisateur des schémas sensori-moteurs. Ceci passe l'utilisation d'un mode d'interaction par essais/erreurs qui n'existe pas dans le système actuellement.

Pour le système, la possibilité de réaliser de tels interactions (par des annotations) nécessite au préalable d'avoir une couche réflexive fonctionnelle qui lui permette d'estimer la qualité de son interprétation propre pour éventuellement revenir sur ces annotations (*cf.* section A.1).

Pour l'utilisateur, l'idée est de lui donner la possibilité d'*annuler* des annotations. L'annulation d'une annotation permettrait de faire des retours en arrière sur l'annotation en cours et sur ses conséquences (*i.e.* les traitements réalisés par le système en réaction à son annotation). L'annulation serait alors une commodité de l'interface qui permet à un utilisateur de corriger une erreur de manipulation (comme un `Ctrl+Z` dans un éditeur quelconque sur PC) ou bien de faire des essais d'annotations de sorte à voir la réaction du système (et ainsi mieux appréhender son fonctionnement) et d'ajuster ces annotations en fonction de ce qu'il veut faire apprendre au système. Au fur et à mesure de l'utilisation et avec une meilleure connaissance du système, l'utilisateur pourrait plus facilement lui transmettre les connaissances qu'il souhaite.

A.4 Choix des annotations comme moyen de communication

Ceci nous amène à discuter le choix des annotations comme moyen d'interaction entre le système et l'utilisateur. La notion d'annotation est une idée forte pour permettre de convertir l'idée de la communication par l'environnement dans le cas de la réalisation d'une interprétation.

Pour la conception de l'annotation comme moyen de communication, nous avons poussé au bout l'idée de la communication par des éléments sémantiquement communs aux deux acteurs de l'interprétation. Ceci a été justifié par l'envie d'assurer une plus grande autonomie du système par rapport à l'utilisateur. Dans les expérimentations, cet objectif est parfois atteint puisque l'utilisation d'annotations permet de faire rentrer certains utilisateurs dans leur tâche d'interprétation sans qu'ils se soucient de l'importance qu'elles ont pour le système. En revanche, de manière pratique les annotations que nous utilisons pour assurer la compréhension commune des annotations sont assez peu expressives au regard de ce qui peut être utilisé par des cliniciens (Nous avons pu rencontrer cela au travers des travaux sur le système MIMIC [178], ainsi que pour la thèse de Bringay [38]). L'ajout d'un contenu textuel apporterait à l'utilisateur une utilité pratique indéniable. Cela lui permettrait d'ancrer durablement ses réflexions dans les données. Cependant, ces contenus textuels ne seraient pas utilisés par le système, on remettrait alors en cause notre principe de symétrie entre les deux agents.

Sans même remettre en question ce principe de symétrie de la compréhension de l'ensemble des annotations, des "contre-annotations" pourraient être ajoutées aux annotations possibles actuellement dans notre système. Il nous paraît important de pouvoir proposer des annotations qui seraient en fait

des contre-exemples. Et de même qu'il existe des mécanismes d'adaptation aux annotations positives, il faudrait adapter les mêmes mécanismes aux annotations négatives. Par exemple, dans le cas de la classification, nous avons montré que le système pouvait prendre en compte des liens entre exemples, il faudrait alors pouvoir créer des anti-liens (deux exemples qui ne peuvent jamais appartenir à une même classe) pour les prendre en compte dans le système.

En résumé, la tentative d'opérationnalisation du concept de l'Enaction nous a conduit à proposer un système répondant aux concepts mais dont les fonctionnalités pratiques sont à améliorer. On peut en particulier mettre en évidence la nécessité de donner la capacité au système d'analyser ses propres traces de fonctionnement dans le but 1) de pouvoir donner à l'utilisateur des explications de ses actions, 2) planifier ses actions et 3) assurer la continuité entre les modifications internes constituant l'interprétation et les connaissances externes apportées par l'utilisateur.

La voie de réflexion qui nous semble intéressante est celle de la construction de mécanismes de base fonctionnant sur le modèle d'un système autopoïétique (comme l'algorithme MN que nous proposons) et sur lequel on pourrait greffer des fonctionnalités réflexives. Le fonctionnement autopoïétique assurerait l'adaptation, l'autonomie et l'unité du système pour réaliser les différentes fonctionnalités de la manière souhaitée par l'utilisateur qui l'aura personnalisé par son usage.

Nous pensons que la perspective ouverte par la conception d'un système sur la base d'un système Enactif, autopoïétique est à poursuivre car elle offre un cadre de réflexion intéressant pour concevoir des systèmes capables de s'adapter et s'individualiser à chaque utilisateur. Mais cette réflexion doit se faire en prenant compte des réalités de réalisation de tels systèmes. Car la réalisation d'un système sur les principes de l'Enaction est confrontée à des verrous difficiles à lever.

B Réalisation d'un système sur les principes de l'Enaction

L'ambitieux objectif de construction d'un système pour l'interprétation collaborative de séries temporelles sur les principes de l'Enaction nous a confronté à des verrous qu'il a fallu lever au moins en partie pour la réalisation technique d'un système sur ces principes. Ces verrous génériques sont :

- l'adaptation des traitements au cours de l'*expérience* du système,
- l'autonomie de traitement se basant sur la construction de la *cohérence* des éléments de système.

Les seules évaluations de la faisabilité d'un tel système sont déjà encourageant car il montre la possibilité du développement de système sur ces principes. Dans le cadre du développement de tel système, nous pouvons mettre en avant la place de l'algorithme de classification. Il est une brique essentielle à la dynamique d'ensemble du système.

Cependant, il reste des améliorations à amener pour s'approcher plus des principes de conceptions que nous avons rencontré au fur et à mesure des chapitres. Pour la tâche d'interprétation nous pouvons revenir sur l'expressivité qu'apporte l'interprétation construite ensuite sur le problème de la complexité calculatoire (étroitement lié à l'expressivité recherchée) et enfin sur la difficulté introduite par la redondance des traitements dans les triades intermédiaires.

B.1 Expressivité de l'interprétation

Dans ce travail, une approche conceptuelle de la construction du sens nous a permis de dégager trois dimensions sémantiques élémentaires des séries temporelles : les segments, les événements et les scénarios.

Nous identifions trois améliorations possibles de l'expressivité de notre système :

- l'ajout pratique de la dimension contextuelle des données qui n'a pas été prise en compte,
- la prise en compte de la granularité temporelle,
- la construction d'explications focalisant mieux les événements caractéristiques *expliquant* les occurrences d'un événement.

B.1.1 Ajouter la dimension contextuelle

Notre interprétation ne permet pas de prendre en compte le contexte d'enregistrement d'un patient. Dans la pratique, le contexte peut être choisi par la restriction des données qui sont à interpréter. Mais à terme, il peut être intéressant d'arriver à introduire le contexte comme une nouvelle dimension de l'interprétation par laquelle les modèles d'évènements et de scénarios seraient spécifiés, mais cette dimension permettrait également de construire et découvrir des contextes communs de patients¹ à partir d'un ensemble large de données (*i.e.* dans lequel plusieurs contextes ou groupes de patients pourraient être identifiés). Les contextes pourraient ainsi permettre de co-déterminer des groupes de patients ainsi que les scénarios et les évènements qui les caractérisent.

Par rapport au schéma conceptuel de l'interprétation de la figure 2.4 page 50, un niveau d'abstraction supplémentaire permettrait de construire des groupes de patients à partir de *modèles* décrivant les caractéristiques de ces groupes. Par rapport au schéma de l'opérationnalisation (figure 2.6 page 51), une triade englobant tous les traitements pourrait représenter la co-détermination des interprétations et des groupes de patients.

B.1.2 Problème de la granularité temporelle

Notre système permet de réaliser des interprétations à l'aide de segments ou symboles qui ont des tailles différentes, et la notion de scénario permettant de regrouper des évènements sous la forme de scénarios, mais nous nous sommes limités à ces deux seuls niveaux : le problème difficile de la prise en compte la granularité n'a pas été explicitement abordé dans notre système.

Pour prendre en compte la granularité, nous pouvons élargir notre définition de scénario comme ensemble de symboles, à celle d'un ensemble de symboles, segments et scénarios (un scénario pouvant contenir des scénarios de granularité temporelle inférieure). Le schéma conceptuel de l'interprétation devient alors un schéma récursif². Le schéma résultant de l'opérationnalisation en triades pourrait alors ressembler à celui de la figure 5.1.

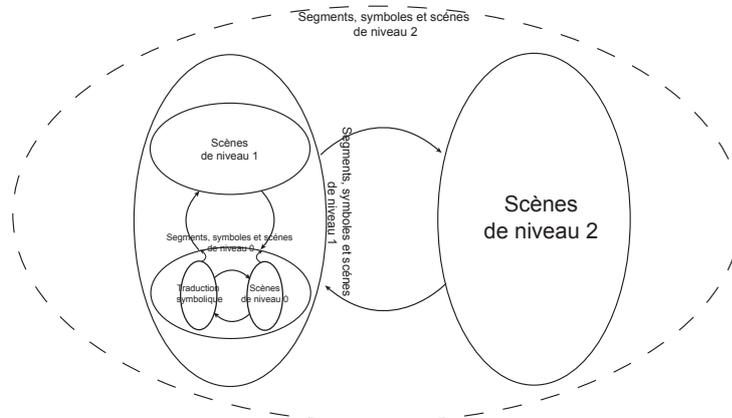


FIG. 5.1.

Architecture récursive de la construction d'une interprétation permettant de prendre en compte la granularité temporelle dans l'interprétation. On ne détaille pas l'intérieur des triades qui est constitué de triade réalisant l'une l'annotation et l'autre la construction de modèles. Dans ce schéma, la notion de scène est étendue à des scènes pouvant inclure des scènes de granularité temporelle inférieure.

Cependant, l'introduction de la granularité nécessiterait de lever deux verrous importants :

1. la mise en place des mécanismes génériques permettant d'assurer des liens entre les niveaux de granularité.
2. la distinction entre changement de granularité et variabilité des durées des évènements.

¹Le caractérisation des informations que doit contenir le "contexte" (ou la caractérisation de "groupes de patients") serait en soit un travail important, on peut rapidement citer quelques éléments qu'il devrait contenir : caractéristiques du patient, de ses symptômes et de sa pathologie ...

²On peut noter que l'introduction de la récursivité nous rapproche de la conception d'un système selon le principe des hiérarchies enchevêtrées.

Pour le second verrou, nous avons effectivement introduit la possibilité de tenir compte de tailles variables pour les événements construits par les *agents de classifications*, la question qui est alors posée est : comment permettre au système de distinguer les différences qui tiennent de la variabilité entre les durées d'événements et celles qui dépendent de la granularité ? Cette question semble difficile.

B.1.3 Manque de focalisation dans la construction des explications

L'utilisation de la supervision de l'apprentissage de scénarios à l'aide de la notion d'explication permet de superviser l'apprentissage en cherchant des points communs dans des exemples positifs du scénario. Cependant, lorsque les symboles d'un événement sont très fréquents ce symbole va servir d'explication à l'occurrence de tous les événements. Plus généralement, des parties entières d'explications peuvent être communes à deux événements. Par exemple, si l'explication d'un événement A est constitué de B et C avec C avant B , alors l'explication de B sera C . En effet, dans la mesure où les motif ABC sont fréquents, le motifs BC le sera également.

Or trouver une explication qui ne permet de focaliser sur la partie caractéristique qui expliquerait l'occurrence d'un événement fait perdre de l'intérêt aux scénarios construits. Les éléments saillants n'étant pas bien identifiés, le sens qui pourrait être attribué au scénario se perd. Il serait plus intéressant d'être capable de fournir uniquement des explications constituées d'événements qui soient exclusivement attribués à une seule explication.

Dans le cadre des données applicatives de patients ventilés. Ce problème a conduit à la construction de scénarios sans grand intérêt. En effet, du fait de la périodicité de la série, un symbole s'*explique* par les symboles du même cycle.

Ce problème est lié au fait que les *agents de construction de scénarios* sont totalement découplés les uns des autres. En particulier ils ne se communiquent pas les événements qu'ils utilisent pour leur explication. Il nous semble que l'introduction de coopération entre les *agents de construction de scénarios* est une perspective pour obtenir des résultats intéressants par l'apprentissage d'explications.

Coopération entre les *agents de construction de scénarios* Une coopération entre les *agents de construction de scénarios* permet d'envisager l'apprentissage des scénarios à l'image de la construction des événements, *i.e.* par une auto-organisation des *agents de construction de scénarios* et par leur intermédiaire l'auto-organisation des scénarios.

Deux directions peuvent être envisagées :

- Les agents doivent communiquer entre eux. Il s'agit par exemple d'avoir connaissance des inconsistances entre les scénarios appris ou de pouvoir fusionner des informations.
- Les agents doivent être créés dynamiquement. Des agents doivent pouvoir être créés dynamiquement par d'autres agents de scénarios pour des besoins de répartition des tâches.

Par leurs interactions, les agents forment ainsi un réseau de contraintes sur les consistances entre les scénarios. L'apprentissage qu'ils mènent dépend de ce réseau, et son résultat constitue ce réseau. On se trouve ici dans une situation d'auto-organisation des agents d'où émergera le résultat global.

B.2 Complexité des algorithmiques

La possibilité qu'offre notre système en terme d'expressivité de l'interprétation produite est à contrebalancer avec la complexité calculatoire pour les produire. Les traitements qui sont réalisés sont globalement complexes. De plus, nous nous ne sommes pas imposés de contrainte de temps de calcul pour la réalisation du système, mais c'est au final une limite pratique du système si on veut envisager de l'utiliser sur une grande masse de données.

B.2.1 Estimation de la complexité des algorithmes

Nous nous intéressons à la complexité calculatoire pour être en mesure d'estimer le temps de traitement de nos algorithmes³.

³Nous ne nous sommes pas intéressés à la place mémoire nécessaire.

Segmentation des séries temporelles Le fait de vouloir tenir compte de modèles d'évènements avec des tailles différentes rend la segmentation des séries temporelles très coûteuse en temps. Si une série temporelle est de longueur L et que le modèle recherche des segments d'une longueur entre l_{min} et l_{max} . Alors la complexité algorithmique pour trouver un segment est proportionnelle à la taille de la matrice de similarité : $(L - 2.l_{min}) \cdot (l_{max} - l_{min})$.

De plus, chacun des N *agents de segmentation* réalise indépendamment sa segmentation, par conséquent la segmentation d'une série temporelle et de l'ordre de :

$$(L - 2.l_{min}) \cdot (l_{max} - l_{min}) \cdot N.NbreSeg,$$

où *NbreSeg* est le nombre de segments découverts par un *agent de segmentation*.

L et $(l_{max} - l_{min})$ pouvant être grand, le temps de calcul d'une segmentation d'une série temporelle peut être très important. Pour nos données de patients ventilés $L \approx 15000$, $(l_{max} - l_{min}) \approx 250$ et le temps de segmentation est de l'ordre de la minute.

Il est très clair qu'un tel algorithme est très coûteux en temps. Les bons résultats qui sont obtenus pour cet algorithme le sont à ce prix. Sans l'adaptation des tailles des fenêtres, les temps de calcul sont bien meilleurs, mais il est alors impossible d'envisager d'avoir des reconnaissances de cycles respiratoires inter-patients, car leurs tailles sont différentes.

Classification L'algorithme de classification étant incrémental, la complexité est au moins proportionnelle au nombre d'exemples à classer. Même si théoriquement les perturbations et les cycles peuvent conduire à un temps de calcul infini, l'implémentation bride les comportements cycliques. En conséquence, les temps de calcul sont raisonnables et en tout cas négligeables par rapport aux temps de segmentations des séries temporelles.

Construction de scénario et fléau de la dimension L'implémentation de l'algorithme *APriori* semblable à celui de C. Dousson est une implémentation qui ne construit pas les contraintes quantitatives, mais sélectionne uniquement les contraintes les plus larges. La complexité qui en résulte est celle de l'algorithme *APriori*. Avec les seuils de fréquence assez élevés, l'élagage rapide des solutions invalides permet d'avoir des temps de calcul courts. À terme, on pourra être tenté de réduire la valeur du seuil de fréquence ce qui peut faire croître exponentiellement les temps de calcul (en n'élaguant plus les branches, on se retrouve à explorer l'ensemble de l'espace des scénarios possibles). L'utilisation des optimisations de l'algorithme *APriori* peut alors être envisagée pour rester dans des temps de calculs raisonnables.

Cependant, avec un tel algorithme, on a pu voir dans les résultats pratiques (*cf.* section D du chapitre 4) que les contraintes temporelles extraites étaient peu informatives. Il faut donc certainement se pencher vers des solutions qui intègrent pleinement la dimension temporelle dans leur apprentissage. L'apprentissage des contraintes temporelles est un gain important dans la qualité du système par la qualité des *feedback* qui peuvent être introduits.

La construction de scénarios est alors confrontée au problème théorique du fléau de la dimension. Le fléau (ou malédiction) de la dimension de Bellman [22] est un phénomène statistique qui fait intervenir le *phénomène de l'espace vide* et le *peaking phenomenon* (*cf.* la thèse de JF Robineau [177]). Retenons simplement de ces deux phénomènes, qu'il est très difficile de connaître le nombre d'exemples nécessaires pour faire un apprentissage. Par exemple, pour l'approximation d'une distribution normale multivariée, le résultat de Silverman [196] montre que la taille du jeu d'exemples doit croître exponentiellement.

Cette limite théorique s'impose à nous comme une contrainte à l'apprentissage d'un scénario. Un scénario et ses contraintes quantitatives (*i.e.* des intervalles de durées sur les évènements et leurs dates) est représentable par un volume dans un espace de dimension égale à la taille du motif (*cf.* Annexe C). Le nombre d'exemples nécessaire pour estimer de manière significative les contraintes temporelles croît exponentiellement avec la taille du motif. Par conséquent, la taille des motifs à apprendre sera théoriquement limitée.

B.2.2 Problème pour la mise en place de schèmes sensori-moteur

La réussite d'un couplage structurel passe par la construction de schèmes sensori-moteur entre les deux systèmes couplés. La mise en place de tels schémas d'interaction nécessite d'avoir une forte réactivité du

système aux actions de l'utilisateur. Or, les traitements massifs qui sont réalisés par notre système est une limite la réactivité du système. Même avec des jeux de données réduits comme le sont ceux de nos jeux de tests, les temps de calculs pour l'exécution des agents peuvent parfois dépasser la minute.

Il semble difficile de réduire cette complexité dont l'origine est la grande expressivité des l'interprétations qui peuvent être produites grâce aux algorithmes choisis. On pourrait effectivement ne pas tenir compte de la dimension temporelle dans les scénarios ou ne pas rechercher des segments de tailles variables, mais alors les résultats de l'interprétation seraient beaucoup moins intéressants.

B.2.3 Perspective d'amélioration des temps de calcul

L'amélioration des temps de calcul est nécessaire pour envisager d'utiliser un tel système pour traiter un grand nombre d'enregistrements de patients. Mais la solution n'est pas dans la réduction de la richesse d'interprétation fournie par les algorithmes. De plus, les séries temporelles multivariées représentent une très grande quantité de données et certains traitements montrent des limites théoriques à l'amélioration des algorithmes. Par conséquent, la tâche d'interprétation nécessitera toujours beaucoup de calculs.

Pour réduire les temps de calculs, la perspective est certainement d'utiliser des moyens de traitement de masse (*e.g.* grille de calculs) comme il est proposé dans le projet ISIS [151]. Ces moyens de traitements peuvent être envisagés de manière structurée en répartissant les données par groupe de patients sur des îlots de calculs.

B.3 Difficulté des traitements des triades intermédiaires

L'utilisation de traitements dans les triades intermédiaires (Triade de symbolisation et Triade Scènes) permet de faciliter les traitements distribués et également l'insertion des perturbations par les interactions de l'utilisateur. Mais ainsi, une partie des traitements sont redondants, par exemple l'attribution des segments à un événement est faite lors de la classification, mais les *agents de symbolisation* réalisent une nouvelle attribution des segments aux événements.

Mais au delà de ce problème de calcul, l'existence de niveaux intermédiaires pose un problème plus important. Celui d'assurer une continuité entre les anciens modèles et les nouveaux modèles⁴. Par exemple, les modèles d'événements sont conservés par l'*agent de symbolisation* et discutés avec l'utilisateur. En particulier, l'utilisateur peut attribuer une appellation clinique à un modèle d'événement. Mais pendant ce temps, les *agents de construction d'événements* construisent de nouveaux événements qui sont reportés dans l'*agent de symbolisation* qui doit alors mettre en correspondance les nouveaux modèles avec les anciens modèles pour y reporter l'appellation. Cette étape est difficile et délicate. Elle est délicate puisque si on effectue une mauvaise correspondance, tous les symboles associés à un événement se retrouvent avec un nom erroné. Elle est difficile puisqu'elle nécessite d'être capable d'établir des correspondances entre les modèles, ce qui sous-entend une mesure de distance entre les modèles. Nous avons déjà insisté sur la difficulté de donner des mesures. Dans l'état actuel d'implémentation de notre système, cette correspondance n'est pas réalisée.

Notre solution serait d'utiliser de nouveau des interactions par des exemples. Plutôt que d'attribuer un nom à un modèle d'événement, il faudrait l'attribuer à un symbole qui serait alors protégé de toute modification par le système. Pour les scénarios, il faudrait également pouvoir fixer des scènes (et donc également fixer tous les symboles qui composent la scène) pour y attribuer un nom.

C Réponse de l'approche au problème applicatif

Nous nous intéressons maintenant à discuter la dimension applicative du travail. En effet, la réalisation du système a été motivé par la difficulté de construction de modèle pour l'amélioration des systèmes de monitoring en unité de soins intensifs. Par notre définition de l'interprétation des séries temporelles, on a pu mettre en évidence que l'apprentissage de modèle était réalisé par la construction d'une interprétation des séries temporelles. On s'appuyait pour cela sur l'idée principale que la construction d'un modèle devait se faire par un cercle vertueux amenant les cliniciens à proposer des modèles, les évaluer hors-ligne

⁴ On peut noter que ce problème de continuité des traitements est un problème bien connu dans le domaine de l'intelligence artificielle.

sur des données réelles connues puis raffiner ces modèles pour qu'ils correspondent aux données. Intégrer ce cercle vertueux avec celui réalisé par le système permet d'ouvrir des perspectives intéressantes.

Bien évidemment dans sa conception actuelle, le système pourrait être difficilement utilisé par des cliniciens. La dimension applicative du système reste finalement assez théorique.

C.1 Difficulté de l'application à des données biomédicales

Les données biomédicales sont des données complexes. En effet, on peut envisager d'utiliser notre système pour l'interprétation de trace de fonctionnement de systèmes techniques comme des procédés chimiques, une centrale nucléaire, le système électronique central d'un avion... avec le même but de construire des modèles d'évènements et de scénarios pour construire des systèmes de monitoring. Mais il y a deux grandes différences entre ces données et des données biomédicales :

- par construction, un système technique a un fonctionnement reproductible, mais des systèmes biomédicaux ont une grande variabilité (intra- et inter- patients)
- les concepteurs sont capables d'indiquer *a priori* quelles traces du système il faut enregistrer pour capturer le fonctionnement du système (même si ces traces sont très volumineuses). Pour des applications biomédicales, il est difficile de capturer les informations saillantes et dans le cadre générale de les identifier.

De plus, nous n'avons pas abordé les problèmes usuels liés au traitement des séries temporelles qui sont particulièrement présents dans les données biomédicales : le bruit, les données manquantes, ou encore la prise en compte de l'incertain naturel des données.

C.1.1 Difficultés rencontrées avec nos données

Au cours de notre travail, nous avons pu explorer des données provenant de patients ventilés (*cf.* section A.3.1 du chapitre 4) et également de patients monitorés en USI par plusieurs appareils (Ventilateur, ECG, SpO_2) sur une durée longue⁵. Elles ont été acquises dans le cadre du projet OSSCAR [53]. On illustre ces données (dénommées par la suite "données OSSCAR") par la figure 5.2.

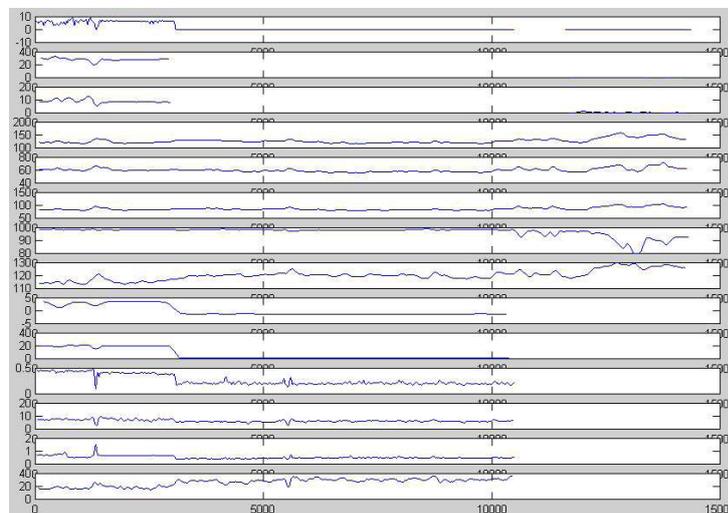


FIG. 5.2.

Exemple de données filtrées provenant d'un patient monitorés pendant 4 heures. Les trois premiers signaux correspondent à des paramètres de ventilation, les trois suivants donnent les différentes pressions sanguines. Le 7ème signal correspond à la SpO_2 , saturation en O_2 , et le 8ème à la fréquence cardiaque. Données acquises par A.S. Silvent dans le cadre du projet OSSCAR [194].

Pour les données de ventilation utilisées dans nos expérimentations, on peut regretter que ces données (cycliques) de cycles respiratoires ne soient pas démonstratives pour le concept de scénario. En revanche,

⁵Les données de patients monitorés en USI ont été acquises et utilisées par A.S. Silvent dans une thèse précédente la présente [194].

elles nous ont permis de mettre en évidence la limite donnée dans la section B.1.3.

Les modèles utilisés n'apportent pas de nouveauté par rapport aux travaux de caractérisation formelle de la forme des cycles respiratoires correspondant à des asynchronies ou non. Mais ce n'était pas là l'objectif de notre travail. Dans la perspective de l'exploration de nouvelles données, il est très satisfaisant d'être capable de retrouver les modèles qui ont une signification clinique. Elle laisse entrevoir la possibilité d'extraire des connaissances intéressantes dans un cadre un peu plus exploratoire.

Pour les données OSSCAR, bien qu'acquises dans le cadre d'un protocole, les situations rencontrées dans les enregistrements sont très variées et il est très difficile pour les cliniciens d'identifier les événements au sein d'un même enregistrement et entre les enregistrements. Nous avons donc ici un cadre réellement exploratoire des signaux physiologiques.

Mais si dans notre situation nous ne pouvons pas conclure c'est principalement du fait d'une faible de quantité de données. Notre système fonctionne sur des principes statistiques et nécessite d'avoir plusieurs occurrences d'un même événement pour pouvoir le caractériser. Si les événements sont peu fréquents, seule une très grande quantité de données peut permettre le fonctionnement de notre système. Le projet ISIS [151] est un projet récent qui s'attache à construire des entrepôts de données de signaux recueillis dans des unités de soins intensifs. Avec de tels quantités de données (2500 heures d'enregistrements multi-paramétrés en seulement 3 mois de fonctionnements) et une réelle capacité calculatoire à les traiter (*i.e.* avec des grilles de calculs), notre système peut espérer apporter une aide à l'exploration de celles-ci.

En conséquence, les données applicatives dont nous disposons pour ce travail et les expérimentations menées sur celles-ci ne permettent pas de mettre pleinement en évidence l'intérêt pratique de notre système. Avant de passer à des traitements sur des grands jeux de données (qui nécessiteraient beaucoup de temps de développement et de mise en place d'un système robuste et efficace), il nous paraît plus judicieux de prolonger la validation de l'approche de l'interprétation collaborative en choisissant des données d'applications mieux adaptées aux capacités du système.

C.1.2 Données médicales d'application du système

La question qui est alors posée est d'identifier les caractéristiques des données biomédicales⁶ pour lesquels notre système présente un intérêt, *i.e.* qu'il permettra aux cliniciens de découvrir de nouveaux modèles d'évènements ou de scénarios.

Les contraintes sur les données sont assez fortes. Les données doivent être à la fois suffisamment complexes pour justifier de l'utilisation du système mais pas trop non plus pour que les méthodes mises en place puissent fonctionner. De plus, il faut que les données répondent aux hypothèses d'existence d'évènements et de scénarios (*cf.* section B.2.1 du chapitre 2).

Pour l'essentiel, les données les plus adaptées à notre système sont celles pour lesquels les notions d'évènements et de scénarios sont supposées identifiables. C'est-à-dire qu'il existe des événements qui peuvent être identifiés dans les enregistrements par des segments caractéristiques (dont la caractérisation est à découvrir) et des scénarios qui sous-tendent l'idée que la dynamique du système est faite d'enchaînements d'évènements.

Ces hypothèses sont fortes si on considère un système biologique en état normal. En effet, on peut se demander si le fonctionnement d'un système biologique est fait d'évolutions lentes, d'évènements brusques ou peut être même des deux à la fois? Des observations cliniques ne trouvent pas nécessairement leur explication dans l'occurrence d'évènements mais dans des modifications lentes qui finissent par provoquer une rupture.

L'utilisation de données acquises dans le cadre d'un protocole permettrait de disposer de données plus formalisées. L'exploration de situations plus cadrées permettrait de contenir la variabilité rencontrée dans les données. L'exploration de tels situations peut permettre de découvrir les traces à la fois de l'évolution correcte du protocole et des dysfonctionnements qui peuvent être rencontrés. Nous avons déjà pointé dans le chapitre 1, les améliorations pratiques que pouvaient apporter des systèmes de monitoring capables d'indiquer l'évolution dans le déroulement d'un protocole et les éventuels déviations.

Les données d'unités de soins intensifs du type des données OSSCAR restent donc les données d'appli-

⁶Nous nous concentrons dans ce travail sur des données biomédicales, mais on pourra faire la transposition des caractéristiques pour des séries temporelles dans des contextes applicatifs différents.

cations privilégiées de notre système. Le système nécessite cependant une plus grande quantité de données pour adresser le problème de la variabilité. On retombe alors sur le même problème que précédemment. La solution passe peut être par l'exploration d'un domaine applicatif médical différent.

Les séries temporelles provenant d'EEG (Electro-Encéphalo-Gramme) ou les EOG (Electro-Occulo-Gramme) sont également des signaux qui nous semblent intéressants à explorer avec notre système. En particulier, dans le cadre de jeunes patients faisant des crises épilepsies, pour certains, il s'agit uniquement de problèmes bénins, mais pour d'autres ces crises peuvent indiquer que le patient est épileptique. Pour réaliser leur diagnostic, les cliniciens sont amenés à réaliser des enregistrements par un EEG de très longues durées. Ils explorent alors ces enregistrements à la recherche d'évènements caractéristiques de l'activité cérébrale : Encoches frontales, Pointe Vertex, Polypointe lennox ... Le diagnostic d'une pathologie peut être fait à partir de la présence de ces évènements.

L'automatisation de l'annotation d'une grande quantité de signaux permettrait de faciliter le travail d'interprétation des cliniciens. L'exploration de la notion de scénarios pourrait également présenter l'intérêt de mieux comprendre l'activité cérébrale du patient.

C.2 Nouvelle perspective du système : Aide à l'annotation

Finalement, le système développé peut être utilisé pour résoudre un problème applicatif plus abordable que celui de l'interprétation mais non moins essentiel : celui de l'aide à l'annotation des séries temporelles. Comme il a été pointé par le biais de la présentation du système MIMIC-II [178] (*cf.* section D.3 du chapitre 1), l'automatisation des annotations représente un réel défi pour les cliniciens.

Le système développé ouvre la perspective d'être valorisé sous la forme d'un logiciel d'aide à l'annotation. Dans un soucis plus pragmatique que celui de la démarche qui nous a conduit à proposer notre approche, il s'agirait de répondre à ce besoin par un système à l'allure plus standard aux systèmes informatiques "classiques", *i.e.* totalement contrôlé par un utilisateur humain.

Tout en conservant l'approche collaborative pour la construction adaptative des modèles, il faut ajouter au système CATS des fonctionnalités pratiques. Certains utilisateurs ne sont pas désireux d'utiliser la construction d'une interprétation par le système impliquant le bouclage des traitements et les modifications des annotations antérieures. Ils cherchent uniquement à utiliser à leur guise les mécanismes d'automatisation des annotations : segmentation, symbolisation ou identification de scènes. Les algorithmes de construction des modèles permettraient de faciliter leur acquisition mais les modèles utilisés pour les annotations automatiques pourraient être choisis et modifiés par l'utilisateur.

N'étant plus limité par l'utilisation d'annotations comme un moyen de communication devant être *compris* par le système informatique, l'annotation symbolique des séries temporelles (*i.e.* au moyen de segments, symboles et scènes) peut être enrichie par des annotations textuelles :

- des enregistrements : en décrivant le contexte d'acquisition des données et le sujet/patient enregistré (sa pathologie, son age, ...),
- des signaux : en précisant par exemple l'instrumentation et les conditions particulières rencontrées pendant l'enregistrement,
- des annotations symboliques elles-mêmes : en donnant la possibilité d'ajouter un contenu textuel associé.

On se rapproche ainsi de la notion d'annotation décrite par Bringay et *al.* [39] pour des dossiers de patients informatisés. On permettrait ainsi de disposer d'un outil tourné vers l'annotation collaborative des données entre cliniciens ou chercheurs, permettant à chacun d'ancrer durablement ses connaissances et ses remarques pour disposer de données de qualité.

Envisagé à long terme, un tel outil peut être un support aussi bien pour la recherche :

- en physiologie : il permet la consultation des données et la constitution d'archives de signaux annotés,
- en traitement de signal : il peut être le support à la conception d'un banc d'essais et d'évaluations d'algorithmes.

Conclusion

L'objectif de cette thèse était d'explorer la conception de systèmes informatiques capables d'œuvrer comme un collaborateur dans la résolution de problèmes complexes. En retour, ces systèmes s'appuient sur leur collaboration avec l'humain pour acquérir des compétences leur permettant à terme de suppléer l'humain dans des tâches complexes.

La contribution de cette thèse à cette question est la proposition d'une approche de la collaboration entre deux agents (le système et l'utilisateur) inspirée du paradigme de l'Enaction de F. Varela. Ce paradigme cognitif et les travaux récents qui en découlent nous ont amené à penser la situation collaborative comme le couplage structurel des agents avec un environnement commun qui sert de support à leur influence mutuelle.

Le bon fonctionnement du couplage structurel découle des propriétés des agents qui y sont impliqués. Selon la paradigme de l'Enaction, ces agents doivent être en particulier :

- autonomes : pour bénéficier pleinement de leurs compétences propres, les agents doivent s'appuyer sur leurs capacités propres. Les compétences ne peuvent pas être directement échangées, mais elles peuvent être acquises par l'expérience. Et si les capacités des agents sont différentes, des voies différentes peuvent être mises en œuvre pour acquérir les mêmes compétences.
- adaptatifs : pour que les systèmes s'enrichissent l'un l'autre pendant la collaboration, ils doivent avoir la capacité d'évoluer pour s'adapter tout au long du couplage. L'adaptation de l'agent lui donne la capacité de s'organiser (se mettre en cohérence) en réaction à une perturbation extérieure.

Pour la conception d'une collaboration homme-machine, nous avons proposé un cadre dans lequel un tel couplage structurel pouvait être envisagé. Tout d'abord, cela a impliqué le choix d'un mode de communication et ensuite un paradigme de conception du système.

Le mode de communication d'un agent dans un couplage structurel doit être une action dans l'environnement. Les modifications qui y sont réalisées seront alors perçues comme des perturbations poussant le système collaborant à s'adapter pour y répondre. Les modifications doivent pouvoir être réalisées à égalité par les deux agents. Nous avons donc proposé comme mode de communication, la communication au moyen d'annotations, c'est-à-dire l'ajout de marques symboliques dans l'environnement qui font elles-mêmes partie de l'environnement.

Le paradigme de conception du système est le paradigme multi-agents qui offre des propriétés intéressantes pour la conception de systèmes autonomes et plastiques.

Cette thèse a eu comme domaine d'application celui d'aider les cliniciens à la réalisation d'une interprétation de séries temporelles multivariées. L'objectif de ces interprétations était de permettre aux cliniciens d'explorer hors-ligne des enregistrements de patients en unités de réanimation de sorte à pouvoir identifier des événements et des scénarios qui soient intéressants par la suite pour améliorer des systèmes de monitoring.

Conclusion

L'interprétation a été présentée comme un processus émergent d'élucidation du "sens" des données. Le "sens" est acquis à partir des éléments saillants des séries temporelles (des segments qui semblent significatifs) selon les dimensions suivantes :

- paradigmatiques : l'identification d'évènements significatif et caractéristiques,
- syntagmatiques : l'identification de scénarios liant localement les évènements,
- contextuelles : le contexte de l'enregistrement d'un patient.

Il découle de cette analyse de l'interprétation un schéma permettant d'opérationnaliser une forme d'interprétation par un système automatique faisant émerger en parallèle les modèles d'évènements et de scénarios ainsi que l'annotation des séries à interpréter.

Le système CATS (*Collaborative Annotation of Time Series*) a été conçu comme un système multi-agents opérationnalisant ces principes dans le contexte d'une application à des signaux physiologiques de patients en réanimation. Ce système est capable de :

- réaliser de manière autonome une interprétation des séries temporelles,
- communiquer par le biais des annotations des séries temporelles,
- adapter son interprétation par le biais du couplage avec l'utilisateur.

Les évaluations qui ont été faites de ce système et de la collaboration nous ont amené à considérer la faisabilité et la pertinence de la conception d'un système collaboratif sur les principes d'Enaction et d'annotation pour l'interprétation de séries temporelles.

Les évaluations se sont attachées à montrer la faisabilité de l'approche, et principalement la faisabilité d'un système qui soit capable d'une adaptation lui permettant d'évoluer au cours d'une collaboration. Les algorithmes et méthodes qui ont été mis en place dans cette thèse commencent à répondre à cette question. Les adaptations qui sont possibles avec notre architecture sont en effet une base pour la construction d'un tel système.

L'approche Enactive se montre pertinente pour mettre en place une situation collaborative. Le système conçu incite les utilisateurs à l'exploration. Mais nous avons mis en évidence que le système manque d'une "couche" pragmatique qui donnerait aux utilisateurs une perception d'un système plus rationnel (et capable d'un fonctionnement intentionnel).

Dans le domaine applicatif de l'interprétation de signaux physiologiques de patients en unité de soins intensifs, les premiers résultats du système montrent un potentiel intéressant pour aider l'interprétation des cliniciens. Une évaluation complète de l'utilité pratique du système nécessiterait d'une part l'implémentation d'un prototype pleinement fonctionnel (*i.e.* avec des temps de calcul plus raisonnables et des interfaces mieux conçues) et d'autre part d'utiliser des données pour lesquelles la recherche de traces d'évènements et de scénarios est intéressante.

De plus, les applications à des données de réanimation sont réputées difficiles. On peut juger positivement les méthodes algorithmiques qui ont été mises en place pendant cette thèse. En particulier, l'apprentissage de scénarios ouvre des perspectives intéressantes pour améliorer les méthodes qui ont été utilisées dans notre implémentation.

Ce travail nous permet d'envisager des applications larges de conception de systèmes centrés sur l'utilisateur en s'inspirant du paradigme de l'Enaction. Ces systèmes acquerraient leurs compétences par l'utilisation que l'homme en ferait et se personnaliseraient de manière transparente pour devenir un collaborateur personnel. L'inspiration de l'Enaction introduit un changement radical dans la conception d'un système. Il ne s'agit plus de concevoir un tel système dans un but donné, mais de le concevoir comme une plate-forme se construisant par son utilisation.

Bibliographie

- [1] O. ABDALA, G. CLIFFORD, M. SAEED, A. REISNER, G. MOODY, I. HENRY et R. MARK : The annotation station : an open-source technology for annotating large biomedical databases. *In Computers in Cardiology, 2004*, p. 681–684, 19-22 Sept. 2004.
- [2] K.-P. ADLASSNIG, C. COMBI, A. K. DAS, E. T. KERAVNOU et G. POZZI : Temporal representation and reasoning in medicine : research directions and challenges. *Artificial Intelligence in Medicine*, 38(2):101–113, 2006.
- [3] R. AGRAWAL et R. SRIKANT : Fast algorithms for mining association rules. *In J. B. BOCCA, M. JARKE et C. ZANILOLO, édés : Proceedings of the 20th International Conference in Very Large Databases*, p. 487–499. Morgan Kaufmann, 1994.
- [4] W. AIGNER, S. MIKSCH, W. MULLER, H. SCHUMANN et C. TOMINSKI : Visualizing time-oriented data - a systematic view. *In Proceedings of Computers and Graphics, 2007*.
- [5] E. AITHOCINE, P.-Y. GUMERY, S. MEIGNEN, L. HEYER, Y. LAVAUT et S. GOTTFRIED : Contribution to structural intensity tool : Application to the cancellation of ECG interference in diaphragmatic EMG. *In Engineering in Medicine and Biology Society, 2006. EMBS '06. 28th Annual International Conference of the IEEE*, p. 5–8, 2006.
- [6] J. ALLEN : Towards a general theory of action and time. *Artificial Intelligence*, 23:123–154, 1984.
- [7] R. AMALBERTI et C. VALOT : Le Magicien d'Oz. Rap. tech., CERMA, 1992.
- [8] E. G. ARIAS, H. EDEN, G. FISCHER, A. GORMAN et E. SCHARFF : Transcending the individual human mind-creating shared understanding through collaborative design. *Transactions on Computer Human Interaction*, 7(1):84–113, 2000.
- [9] F. ARMETTA, S. HASSAS, S. PIMONT et O. LEFEVRE : Vers le contrôle de l'émergence par la coordination de l'activité décentralisée d'agents pour le partage de ressources critiques. *Revue d'Intelligence Artificielle*, 21:A para 4, 2007. Numéro spécial "Modèles multi-agents pour des environnements complexe".
- [10] S. AUBRY, I. THOUVENIN, D. LENNE et S. OKAWA : Knowledge integration for annotating in virtual environments. *International Journal of Product Development*, 4(6):533–546, 2007.
- [11] J. C. AUGUSTO : Temporal reasoning for decision support in medicine. *Artificial Intelligence in Medicine*, 33(2):1–24, 2005.
- [12] J. AUSTIN : *How To Do Things With Words*. Oxford University Press, 1962.
- [13] R. K. AVENT et J. D. CHARLTON : A critical review of trend-detection methodologies for biomedical monitoring systems. *Critical Review of Biomedical Engineering*, 17(6):621–659, 1990.
- [14] P. BACH-Y-RITA, C. C. COLLINS, F. SAUDERS, B. WHITE et L. SCADDEN : Vision substitution by tactile image projection. *Nature*, 221:963–964, 1969.
- [15] B. BACHIMONT : Bibliothèques numériques audiovisuelles : des enjeux scientifiques et techniques. *Document numérique*, 2-3(4):219–242, 1998.
- [16] B. BACHIMONT : Dossier patient et lecture hypertextuelle. problématique et discussion. *Les cahiers du numérique*, 2(2), 2002.
- [17] B. BACHIMONT : *Arts et sciences du numérique : Ingénierie des connaissances et critique de la raison computationnelle*. HDR, Université Technologique de Compiègne, 2004.
- [18] R. BADE, S. SCHLECHTWEIG et S. MIKSCH : Connecting time-oriented data and information to a coherent interactive visualization. *In Proceedings of CHI'04, ACM Conference on Human Factors in Computing Systems*, p. 105–112. ACM Press, 2004.
- [19] M. BASSEVILLE et M.-O. CORDIER : Surveillance et diagnostic de systèmes dynamiques : approches complémentaires du traitement de signal et de l'intelligence artificielle. Rap. tech., INRIA - Rennes, Avril 1996.

- [20] R. BELLAZZI, C. LARIZZA, P. MAGNI, S. MONTANI et G. D. NICOLAO : Intelligent analysis of clinical time series by combining structural filtering and temporal abstractions. *In AIMDM '99 : Proceedings of the Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making*, vol. 1620 de LNCS, p. 261–270. Springer-Verlag, 1999.
- [21] R. BELLAZZI, C. LARIZZA, P. MAGNI, S. MONTANI et M. STEFANELLI : Intelligent analysis of clinical time series : an application in the diabetes mellitus domain. *Artificial Intelligence in Medicine*, 20(1):37–57, 2000.
- [22] R. E. BELLMAN : *Adaptative Control Processes : A Guided Tour*. Princeton University Press, New Jersey, U.S.A., 1961.
- [23] R. BENKIRANE, éd. *La complexité, vertiges et promesses*. Le Pommier (Réédition), 2006.
- [24] A. BERGER et S. PESTY : Towards a conversational language for artificial agents in mixed community. *In Fourth International Central and Eastern European Conference on Multi-Agent Systems CEEMAS'05*, p. 31–40. Springer Verlag, Budapest, 2005.
- [25] C. BERNON, M.-P. GLEIZES, S. PEYRUQUEOU et G. PICARD : *Engineering Societies in the Agents World III : Third International Workshop, ESAW 2002, Madrid, Spain, September 16-17, 2002. Revised Papers*, chap. ADELFE : A Methodology for Adaptive Multi-agent Systems Engineering. LNCS. 70-81, springer berlin / heidelberg édn, 2002.
- [26] L. BIOT, P. Y. CARRY, J. P. PERDRIX, A. EBERHARD et P. BACONNIER : Clinical evaluation of alarm efficiency in intensive care. *Annales Francaises d'Anesthesie et de Reanimation*, 19(6):459–466, 2000.
- [27] L. BIOT, L. HOLZAPFEL, G. BECQ, C. MÉLOT et P. BACONNIER : Do we need a systematic activation of alarm soundings for blood pressure monitoring for the safety of ICU patients? *Critical Care*, 18(4):212–216, 2003.
- [28] J. B. BLACK, S. P. ROBERTSON et W. ZACHARY : *Cognition, Computing, and Cooperation*. LNCS. Intellect, 1990.
- [29] H. BLOCKEEL et L. D. RAEDT : Inductive Logic Programming : Theory and Methods. *The Journal of Logic Programming*, 19/20:629–680, 1994.
- [30] D. BOAZ et Y. SHAHAR : A framework for distributed mediation of temporal-abstraction queries to clinical databases. *Artificial Intelligence in Medicine*, 34(1):3–24, 2005.
- [31] P. BOTTONI, C. GARBAY, F. LECCA, P. MUSSIO et P. RIZZO : Collaborative indexing and retrieval by annotation : the case of artistic artifacts. *In Proceedings of the 2nd International Workshop on Content-based Multimedia Indexing*, p. 315–322, 2001.
- [32] J. BOUJUT et J. DUGDALE : Design of a 3D annotation tool for supporting evaluation activities in engineering design. *In Proceedings of COOP '06 : Seventh International Conference on the Design of Cooperative Systems*, 2006.
- [33] P. BOURGINE et J. STEWART : Autopoiesis and cognition. *Artificial Intelligence*, 10:327–345, 2004.
- [34] R. BOVA, S. HASSAS et S. BENBERNOU : An immune system-inspired approach for composite web services reuse. *In Proceedings de l'atelier "Conception des systèmes d'information et services Web" (INFORSID 06)*, 2006.
- [35] X. BOYEN, N. FRIEDMAN et D. KOLLER : Discovering the hidden structure of complex dynamic systems. *In Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence*, p. 91–100. San Francisco : Morgan Kaufmann, August 1999.
- [36] O. BRDICZKA, P. REIGNIER et J. L. CROWLEY : Detecting individual activities from video in a smart home. *In Proceedings of 11th International Conference on Knowledge-Based & Intelligent Information & Engineering Systems (KES)*, LNCS. Springer, 2007. IN PRESS.
- [37] P. BRÉZILLON : *Encyclopedia of E-Collaboration*, chap. Context-based explanations for e-collaboration. 2008.
- [38] S. BRINGAY : *Les annotations pour supporter la collaboration dans le dossier patient électronique*. Thèse de doctorat, Université de Picardie Jules Verne - Amiens, 2006.
- [39] S. BRINGAY, C. BARRY et J. CHARLET : *Annotations dans les documents pour l'action*, chap. Un modèle pour les annotations du dossier patient informatisé, p. À paraître. Hermes, 2006.
- [40] S. BRINGAY, C. BARRY et J. CHARLET : Annotations for the collaboration of the health professionals. *In Proceedings of AMIA Annual Symposium*, p. 91–95, 2006.
- [41] J. BRUNER : *Toward a theory of instruction*. Harvard University Press, 1966.
- [42] V. BRUSONI, L. CONSOLE, P. TEREZIANI et D. DUPRE : A spectrum of definitions for temporal model-based diagnosis. *Artificial Intelligence*, 102(1):39–79, 1998.

- [43] V. BULITKO et D. C. WILKINS : Machine learning for time interval petri nets. In *Proceedings of the 18th Australian Joint Conference on Artificial Intelligence*, LNAI, p. 959–965, Sydney, Australia, 2005. Springer-Verlag.
- [44] P. BUONO, A. ARIS, C. PLAISANT, A. KHELLA et B. SHNEIDERMAN : Interactive pattern search in time series. In R. F. ERBACHER, J. C. ROBERTS, M. T. GROHN et K. BORNER, eds : *Proceedings of SPIE Conference on Visualization and Data Analysis, VDA 2005*, vol. 5669, p. 175–186, October 2005.
- [45] P. BUONO, C. PLAISANT, A. SIMEONE, A. ARIS, B. SHNEIDERMAN, G. SHMUELI et W. JANK : Similarity-based forecasting with simultaneous previews : A river plot interface for time series forecasting. In *Proceedings of the 11th International Conference on Information Visualisation*, 2007.
- [46] D. CALVELO, M.-C. CHAMBRIN, D. POMORSKI et P. RAVAUX : Towards symbolization using data-driven extraction of local trends for ICU monitoring. *Artificial Intelligence in Medicine*, 19(3):203–223, 2000.
- [47] D. CALVELO, M.-C. CHAMBRIN, D. POMORSKI et P. RAVAUX : Towards symbolization using data-driven extraction of local trends for ICU monitoring. *Artificial Intelligence in Medicine*, 19(3):203–223, 2000.
- [48] J.-C. CAMPAGNE et A. CARDON : Artificial emotions for robots using massive multi-agent systems. In *Proceedings on Social Intelligence Design International Conference*, 2003.
- [49] G. CARRAULT, M.-O. CORDIER, R. QUINIOU, M. GARREAU, J.-J. BELLANGER et A. BARDOU : A model-based approach for learning to identify cardiac arrhythmias. In *AIMDM'99 : Artificial Intelligence in Medicine and Medical Decision Making*, vol. 1620, p. 165–174, Aalborg, Denmark, june 1999. Springer Verlag.
- [50] B. CHAIB-DRAA et D. VANDERVEKEN : Agent communication language : A semantics based on the success, satisfaction and recursion. In *Proceedings of ATAL'98*, 1998.
- [51] M.-C. CHAMBRIN : Alarms in the intensive care unit : how can the number of false alarms be reduced? *Critical Care*, 5(4):184–188, 2001.
- [52] M.-C. CHAMBRIN, S. CHARBONNIER, S. SHARSHAR, G. BECQ et L. BADJI : Automatic characterization of events on SpO_2 signal : comparison of two methods. In *26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, p. 3474–3477, 2004.
- [53] M.-C. CHAMBRIN, J. MANGALABOYI, S. SHARSHAR, P.-Y. CARRY, J.-P. PERDRIX, P. RAVAIX, C. VILHELM, E. BJAOUI, M. DOJAT, C. GARBAY, A.-S. SILVENT et S. CHARBONNIER : OSSCAR : A collaborative project for intelligent patient monitoring in intensive care and anesthesia. In *Proceedings of IDAMAP : Intelligent Data Analysis in Medicine and Pharmacology*, 2003.
- [54] M.-C. CHAMBRIN, P. RAVAUX, D. CALVELO-AROS, A. JABORSKA, C. CHOPIN et B. BONIFACE : Multicentric study of monitoring alarms in the adult Intensive Care Unit (ICU) : A descriptive analysis. *Intensive Care Medicine*, 25(12):1360–1366, 1999.
- [55] J.-P. CHANGEUX : *L'homme de vérité*. Odile Jacob, 2002.
- [56] S. CHARBONNIER : On-line extraction of successive temporal sequences from ICU high-frequency data for decision support information. In *Proceedings of ninth Conference on Artificial Intelligence in Medicine*, p. 1–10, 2003.
- [57] S. CHARBONNIER, G. BECQ et L. BIOT : On-line segmentation algorithm for continuously monitored data in intensive care units. *IEEE Trans Biomed Eng*, 51(3):484–492, 2004.
- [58] G. CHICOISNE et S. PESTY : Un modèles de conversation mixte pour l'interaction humain/agent. In J. BRIOT, éd. : *Journées Francophones sur les systèmes Multi-Agents*, Hammamet, Tunisie, 2003. Hermes.
- [59] L. CHITTARO et A. MONTANARI : Temporal representation and reasoning in artificial intelligence : Issues and approaches. *Annals of Mathematics and Artificial Intelligence*, 28:47–106, 2000.
- [60] B. CHIU, E. KEOGH et S. LONARDI : Probabilistic discovery of time series motifs. In *the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, p. 493–498, Washington, DC, USA, 2003.
- [61] A. CHOUAKRIA-DOUZAL : *Extension des méthodes d'analyse factorielle à des données de type intervalle*. Thèse de doctorat, Université Paris IX Dauphine, 1998.
- [62] A. CHOUAKRIA-DOUZAL : On the distance measure between time series. In *3rd IASC World Conference on Computational Statistics & Data Analysis*, Cyprus, 2005.
- [63] A. CHOUAKRIA-DOUZAL et P. NAGABHUSHAN : Improved fréchet distance for time series. In *Data Science and classification*, 10th Conference of the International Federation of Classification Societies, p. 13–20. Springer, 2006.
- [64] A. A. CLARKE et M. G. G. SMYTH : A co-operative computer based on the principles of human co-operation. *International Journal on Man-Machine Studies*, 38(1):3–22, 1993.

- [65] J. COLLIER : Supervenience and reduction in biological hierarchies. *Canadian Journal of Philosophy*, 14:209–234, 1988.
- [66] J. D. COLLIER et S. J. MULLER : The dynamical basis of emergence in natural hierarchies. In G. FARRE et T. OKSALA, éd. : *Proceedings of the Conference on Emergence, Complexity, Hierarchy and Organisation (ECHO III)*, 1998.
- [67] L. CONSOLE, D. T. DUPRÉ et P. TORASSO : A theory of diagnosis for incomplete causal models. In *Proceedings of IJCAI'89 : International Joint Conference on Artificial Intelligence*, p. 1311–1317, 1989.
- [68] G. CORTELESSA et A. CESTA : Evaluating mixed-initiative systems : An experimental approach. In *Proceedings of the 16th International Conference on Automated Planning & Scheduling (ICAPS-06)*, 2006.
- [69] P. CRAIG, J. KENNEDY et A. CUMMING : Animated interval scatter-plot views for the exploratory analysis of large scale microarray time-course data. *Information Visualization*, 4(3):149 – 163, 2005.
- [70] G. DAS, D. GUNOPULOS et H. MANNILA : Finding similar time series. In *Principles of Data Mining and Knowledge Discovery*, p. 88–100, 1997.
- [71] K. DAVIDSON et V. BARBER : Electronic monitoring of patients in general wards. *Nursing Standard*, 18(49):42–46, 2004.
- [72] J. DE KLEER, A. MACKWORTH et R. REITER : Characterizing diagnoses and systems. *Artificial Intelligence*, 56(2-3):197–222, 1992. repris dans *Readings in Model-Based Diagnosis*, W. Hamscher, L. Console and J de Kleer (dir.), Morgan Kaufmann, p54,65,1992.
- [73] T. L. DEAN et M. P. WELLMAN : *Planning and control*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1991.
- [74] J. DEGUET, Y. DEMAZEAU et L. MAGNIN : Elements about the emergence issue, a survey of emergence definitions. In *Proceedings of the European Conference on Complex Systems (ECCS)*, Paris, France, 2005.
- [75] S. W. A. DEKKER et D. D. WOODS : MABA-MABA or abracadara ? progress on human-automation coordination. *Cognition, Technology and Work*, 4:240–244, 2002.
- [76] J. DERRIDA : *Paper Machine*. Stanford University Press, 2005.
- [77] S. DESCORPS-DECLÈRE, D. ZIÉBELIN, F. RECHENMANN et A. VIARI : Genepi : a blackboard framework for genome annotation. *BMC Bioinformatics*, 7:450–463, 2007.
- [78] F. DESOBRY, M. DAVY et C. DONCARLI : An online kernel change detection algorithm. *IEEE Transactions on Signal Processing*, 53(8):2961–2974, 2005.
- [79] E. DIDAY et L. BILLARD : From the statistics of data to the statistic of knowledge : Symbolic data analysis. *Journal of the American Statistical Association*, 98(462), 2003.
- [80] D. DIONISI : *Proposition d'une méthodologie d'opérationnalisation informatique de l'approche Enactive de la cognition*. Thèse de doctorat, INSA Rouen, 2006.
- [81] D. DIONISI et J. LABICHE : Enaction et informatique : les enjeux de l'opérationnalisation technologique d'une théorie de la cognition. In *Colloque ARCo*, 2006.
- [82] M. DOJAT, N. RAMAUX et D. FONTAINE : Scenario recognition for temporal reasoning in medical domains. *Artificial Intelligence in Medicine*, 14(1-2):139–155, 1998.
- [83] C. DOUSSON et T. DUONG : Discovering chronicles with numerical time constraints from alarm logs for monitoring dynamic systems. In T. DEAN, éd. : *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, p. 620–626. Morgan Kaufmann, 1999.
- [84] J. DOYLE, I. KOHANE, W. LONG et P. SZOLOVITS : The architecture of MAITA : A tool for monitoring, analysis, and interpretation. Rap. tech., Massachusetts Institute of Technology, Laboratory for Computer Science, 1999.
- [85] S. DREISEITL et M. BINDER : Do physicians value decision support ? A look at the effect of decision support systems on physician opinion. *Artificial Intelligence in Medicine*, 33(1):25–30, 2005.
- [86] J. G. DROYSEN : *Outline of the principles of History*. Howard Fertig, 1967.
- [87] F. DUCHÈNE : *Fusion de données multicapteurs pour un système de télésurveillance médicale de personnes à domicile*. Thèse de doctorat, Université Joseph Fourier, 2004.
- [88] F. DUCHÈNE et C. GARBAY : Apprentissage non supervisé de motifs temporels, multidimensionnels et hétérogènes – application à la télésurveillance médicale. In *Conférence francophone sur l'apprentissage automatique (CAP)*. Presses Universitaires de Grenoble, 2005.
- [89] J.-P. DUPUY : *Aux origines des sciences cognitives*. Édition La découverte, 1999.
- [90] P.-V. DUQUAIRE : Introduction à la pensée de Francisco J. Varela. *Les Cahiers de l'ATP*, 4, 2003. http://alemore.club.fr/Duquaire-Introduction_a_Varela-juill2003.pdf.

- [91] J. EDWORTHY et E. HELLIER : Fewer but better auditory alarms will improve patient safety. *Quality and Safety in Healthcare*, 15:212–215, 2005.
- [92] S. ENGELL, G. FREHSE et E. SCHNEIDER : *Modelling, Analysis, and Design of Hybrid Systems*, vol. 279 de *LNCIS*. Springer, 2002. ISBN 3540438122.
- [93] J. FERBER et O. GUTKNECHT : A meta-model for the analysis and design of organisations in multi-agent systems. In *Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS'98)*, p. 128–135. IEEE Computer Society Press, 1998.
- [94] G. FISCHER : Shared understanding, informed participation, and social creativity - objectives for the next generation of collaborative systems. In *Proceedings of COOP'2000*, 2000.
- [95] J. FODOR : *The Modularity of Mind : An Essay in Faculty Psychology*. MIT Press, 1983.
- [96] W. FRIESDORF, B. BUSS et M. GOBEL : Monitoring alarms - the key to patient's safety in the icu? *Intensive Care Medicine*, 25(12):1350–1352, 1999.
- [97] É. FROMONT, R. QUINIOU et M.-O. CORDIER : Learning rules from multisource data for cardiac monitoring. In S. MIKSCH, J. HUNTER et E. T. KERAVNOU, édés : *Proceedings of the 10th conference on Artificial Intelligence in Medicine*, vol. 3581 de *LNCIS*, p. 484–493. Springer, 2005.
- [98] H.-G. GADAMER : *Truth and Method, 2nd Ed, traduction Anglaise J. Wiensheimer et D. Marshall*. New York : Crossroad, 1989.
- [99] J. GAMPER et W. NEJDL : Abstract temporal diagnosis in medical domains. *Artificial Intelligence in Medicine*, 10(3):209–234, 1997.
- [100] C. GARBAY : Pour une conception distribuée des systèmes de vision. *L'Objet*, 8(4), 2002. Numéro Spécial "Systèmes distribués et connaissances".
- [101] R. GAUDIN et N. NICOLYOYANNIS : An adaptable time warping distance for time series learning. In *5th International Conference on Machine Learning and Applications (ICMLA 06)*, 2006.
- [102] J. J. GIBSON : *The Ecological Approach To Visual Perception*. Lawrence Erlbaum (Réédition 1986), 1979.
- [103] R. GOLDSTONE et J. SON : *Similarity*, chap. Cambridge Handbook of Thinking and Reasoning, p. 13–36. Cambridge University Press, 2004.
- [104] J. GRONDIN : Qu'est ce que l'interprétation? In *Skepsis*, p. 126–131, 2004.
- [105] T. GUYET, C. GARBAY et M. DOJAT : Algorithme d'apprentissage de scénarios à partir de séries symboliques temporelles. In R. QUINIOU et G. HÉBRIL, édés : *Atelier Fouille de Données Temporelles, EGC'06 : Extraction et Gestion de Connaissances*, 2006.
- [106] I. HAIMOWITZ, P. LE et I. KOHANE : Clinical monitoring using regression-based trend templates. *Artificial Intelligence in Medicine*, 7(6):473–496, 1995.
- [107] I. J. HAIMOWITZ et I. S. KOHANE : Managing temporal worlds for medical trend diagnosis. *Artificial Intelligence in Medicine*, 8(3):299–321, 1996.
- [108] D. HART, S. CHU, E. KEOGH et M. PAZZANI : Iterative deepening dynamic time warping for time series. In V. KUMAR, H. MANNILA, R. GROSSMAN, J. HAN et R. MOTWANI, édés : *Proceedings of the Second SIAM International Conference on Data Mining*, 2002.
- [109] V. HAVELANGE, C. LENAY et S. J. : Les représentations : mémoire externe et objets techniques. *Intellectica*, 35:115–131, 2003.
- [110] V. HAYWARD, O. ASTLEY, M. CRUZ-HERNANDEZ, D. GRANT et G. ROBLES-DE-LA-TORRE : Haptic interfaces and devices. *Sensor Review*, 24(1):16–29, 2004.
- [111] R. HELD et A. HEIN : Adaptation of disarranged hand-eye coordination contingent upon re-afferent simulation. *Perceptual-Motor Skills*, 8:87–90, 1958.
- [112] L. HEUTTE, A. NOSARY et T. PAQUET : A multiple agent architecture for handwritten text recognition. *Pattern Recognition*, 37(4):665–674, 2004.
- [113] J. HIMBERG, K. KORPIAHO, H. MANNILA, J. TIKANMAKI et H. TOIVONEN : Time series segmentation for context recognition in mobile devices. In *Proceedings of International Conference on Data Mining*, p. 203–210, 2001.
- [114] J.-M. HOC : From human - machine interaction to human - machine cooperation. *Ergonomics*, 43(7):833–843, 2000.
- [115] D. HOFSTADTER : *Godel, Escher, Bach. Les brins d'une Guirlande Eternelle*. Dunod, 2000.
- [116] HONGRE : L'intertextualité, 2004. http://www.webzinemaker.com/admi/m6/page.php3?num_web=8526&rubr=2&id=169447.

- [117] F. HÖPPNER : Learning dependencies in multivariate time series. *In Proceedings of the ECAI'02 Workshop on Knowledge Discovery in (Spatio-) Temporal Data*, p. 25–31, 2002.
- [118] E. HORVITZ : Principles of mixed-initiative user interfaces. *In Proceedings of CHI'99, ACM SIGCHI Conference on Human Factors in Computing Systems*, p. 159–166. ACM press, 1999.
- [119] J. HUNTER : TSNet - A distributed architecture for time series analysis. *In N. PEEK et C. COMBI, édés : Proceedings of the Workshop on Intelligent Data Analysis in Bio-Medicine and Pharmacology, IDAMAP 2006*, p. 85–92, 2006.
- [120] E. HUTCHINS : *Cognition in the Wild*. MIT Press, Cambridge, MA, 1995.
- [121] M. IMHOFF et S. KUHL : Alarm algorithms in critical care monitoring. *Anesthesia & Analgesia*, 102(5):1525–1537, 2006.
- [122] W. ISER : *The Range of Interpretation*. Columbia University Press, 2000.
- [123] F. ITAKURA : Minimum prediction residual principle applied to speech recognition. *In IEEE Transaction on Acoustics, Speech and Signal Proceedings*, p. 52–72, 1975.
- [124] M. R. JEAN : émergence et SMA. *In Journées Francophones de IAD et des SMA*, p. 323–342, 1997.
- [125] P. JONES et C. MITCHELL : Human-computer cooperative problem solving : theory, design, and evaluation of an intelligent associate system. *IEEE Transactions on Systems, Man and Cybernetics*, 25(7):1039–1053, 1995.
- [126] P.-S. KAM et A. W.-C. FU : Discovering temporal patterns for interval-based events. *In Y. KAMBAYASHI, M. K. MOHANIA et A. M. TJOA, édés : DaWaK 2000 : Data Warehousing and Knowledge Discovery, Second International Conference*, vol. 1874 de LNCS, p. 317–326. Springer, 2000.
- [127] L. KARSENTY et P. BRÉZILLON : Cooperative problem solving and explanation. *Expert Systems With Applications*, 8(4):445–462, 1995.
- [128] E. KEOGH, S. CHU, D. HART et M. PAZZANI : Segmenting time series : A survey and novel approach. *Data Mining in Time Series Databases*, World Scientific Publishing Company, 1993.
- [129] E. KEOGH, M. VLACHOS, M. HADJIELEFThERIOU et D. GUNOPULOS : Indexing multi-dimensional time-series with support for multiple distance measures. *9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, p. 216–225, 2003.
- [130] I. KESTIN, B. MILLER et H. LOCKHARD : Auditory alarms during anaesthesia monitoring. *Anaesthesiology*, 69:106–109, 1988.
- [131] A. KOESTLER : *The Ghost in the Machine*. Arkana Books, 1989.
- [132] R. KOSARA et S. MIKSCH : Visualizing complex notions of time. *In Proceedings of the Conference on Medical Informatics*, p. 211–215, 2001.
- [133] E. M. KOSKI, A. MAKIVIRTA, T. SUKUVAARA et A. KARI : Frequency and reliability of alarms in the monitoring of cardiac postoperative patients. *International Journal of Clinical Monitoring and Computing*, 7(2):129–133, 1990.
- [134] S. LAHLOU : Vers une théorie de l'interprétation en analyse des données textuelles. *In S. BOLASCO, L. LEBART et A. SALEM, édés : Proceedings of JADT 1995 : 3rd International Conference on Statistical Analysis of Textual Data.*, vol. 1, p. 221–228, 1995.
- [135] A. LAW, Y. FREER, J. R. HUNTER, R. LOGIE, N. McINTOSH et J. QUINN : A comparison of graphical and textual presentations of time series data to support medical decision making in the neonatal intensive care unit. *International Journal of Clinical Monitoring and Computing*, 19:183–194, 2005.
- [136] J.-L. LE MOIGNE : *La théorie du système général, Théorie de la modélisation. 4ème Édition*. 2006. ISBN 2130465153.
- [137] R. LENGELLÉ : *Décision et reconnaissance des formes en signal*. Traité IC2 Information - Commande - Communication. 2002.
- [138] P. LERAY : *Apprentissage et diagnostic de systèmes complexes : Réseaux de neurones et réseaux bayésiens. Application à la gestion en temps réel du trafic téléphonique français*. Thèse de doctorat, Université Paris 6, 1998.
- [139] J. LIN, E. J. KEOGH, S. LONARDI, J. P. LANKFORD et D. M. NYSTROM : VizTree : A tool for visually mining and monitoring massive time series databases. *In M. A. NASCIMENTO, M. T. ÖZSU, D. KOSSMANN, R. J. MILLER, J. A. BLAKELEY et K. B. SCHIEFER, édés : Proceedings of the Thirtieth International Conference on Very Large Data Bases*, p. 1269–1272. Morgan Kaufmann, 2004.
- [140] A. LOWE, M. J. HARRISON et R. W. JONES : Diagnostic monitoring in anaesthesia using fuzzy trend templates for matching temporal patterns. *Artificial Intelligence in Medicine*, 16(2):183–199, 1999.

- [141] H. MANNILA, H. TOIVONEN et A. VERKAMO : Discovering frequent episodes in sequences. *In Proceedings of 1st International Conference on Knowledge Discovery and Data Mining (KDD'95)*, p. 210–215, 1995.
- [142] G. P. MARELICH, S. MURIN, F. BATTISTELLA, J. INCIARDI, T. VIERRA et M. ROBY : Protocol weaning of mechanical ventilation in medical and surgical patients by respiratory care practitioners and nurses : effect on weaning time and incidence of ventilator-associated pneumonia. *Chest*, 118(2):459–467, 2000.
- [143] R. MARK : Integrating data, models and reasoning in critical care, brp projet proposal. Rap. tech., Harvard-MIT Division of Health Sciences and Technology, 2004.
- [144] S. B. MARTINS, Y. SHAHAR, M. GALPERIN, H. KAIZER, D. GOREN-BAR, D. MCNAUGHTON, L. V. BASSO et M. K. GOLDSTEIN : Evaluation of KNAVE-II : a tool for intelligent query and exploration of patient data. *Medinfo*, 11(1):648–652, 2004.
- [145] G. MARWELL et D. SCHMITT : *Co-operation - an experimental analysis*. London : Academic Press, 1975.
- [146] F. MASSEGLIA, P. PONCELET et M. TEISSEIRE : Extraction de motifs séquentiels : problèmes et méthodes. *Ingénierie des Systèmes d'Information (ISI)*, 9(3/4):183–210, 2004.
- [147] H. MATURANA et F. VARELA : *Autopoiesis and Cognition*. Dordrecht, Holland, 1980.
- [148] H. MATURANA et F. VARELA : *L'arbre de la connaissance - racines biologiques de la compréhension humaine*. Addison-Wesley, 1994.
- [149] L. MAYET : L'énigme de l'émergence. Hors-Série Science et Avenir, juillet 2005.
- [150] D. MCDERMOTT : A temporal logic for reasoning about processes and plans. *Cognitive Science*, 6:101–155, 1982.
- [151] H. MEHDAOUI, B. SARRAZIN, I. EL ZEIN, L. ALLART, C. VILHELM, S. GUERRA, D. ZITOUNI, M. LEMDANI, R. VALENTINO, A. HERBLAND et P. RAVAUX : ISIS program : a new tool for medical research at the bedside in critical care units. *Critical Care Medicine*, 11(Suppl 2):439, 2007.
- [152] S. MERSMANN et M. DOJAT : SmartCaretm-automated clinical guidelines in critical care. *In R. L. de MANTARAS et L. SAIITA, eds : Proceedings of the 16th European Conference on Artificial Intelligence, ECAI'2004*, p. 745–749, Valencia, Spain, 2004. IOS Press.
- [153] J. MIERMONT : *Recherches sur le langage en psychologie cognitive*, chap. Contextualisations, communication et cognition. Dunod, 1997.
- [154] M. MINSKY : *The Society of Mind*. 1988.
- [155] F. MORA, G. PASSARIELLO, G. CARRAULT et J.-P. LE PICHON : Intelligent patient monitoring and management systems : a review. *IEEE Engineering in Medicine and Biology Magazine*, 12(4):23–33, 1993.
- [156] V. MORET-BONILLO, A. ALONSO-BETANZOS, E. GARCIA-MARTIN, B. GUIJARRO-BERDINAS et M. CABRERO-CANOSE : A temporal strategy for intelligent patient monitoring in the intensive care units. *In Proceedings IEEE Engineering Biology Society*, vol. 15, p. 610–611, 1993.
- [157] R. MOSKOVITCH : Mining time oriented data via temporal abstraction. *In J. HUNTER, éd. : Doctoral Consortium of the Artificial Intelligence in Medicine Conference*, 2007.
- [158] T. D. NGUYEN, S. KAWASAKI et T. B. HO : Discovery of trends and states in irregular medical temporal data. *Discovery Science*, p. 410–417, 2003.
- [159] J. O. NILSESTUEN et K. D. HARGETT : Using ventilator graphics to identify patient-ventilator asynchrony. *Respiratory Care*, 50(2):202–34, 2005.
- [160] N. NITANDA, M. HASEYAMA et H. KITAJIMA : An audio signal segmentation and classification using fuzzy c-means clustering. *In Proceedings of the 2nd International Conference on Information Technology for Application (ICITA 2004)*, 2004.
- [161] P. NORRIS, B. DAWANT et K. SUWANMONGKOL : Improving the simon architecture for critical care intelligent monitoring. *In Proceedings of AMIA Annual Symposium*, p. 794–798, 1997.
- [162] T. M. O'CARROLL : Survey of alarms in an intensive therapy unit. *Anaesthesia*, 41(7):742–744, 1986.
- [163] M. OCCELLO, C. BAEIJS, Y. DEMAZEAU et J.-L. KONING : Mask : An aeio toolbox to design and build multi-agent systems. *In C. et AL., éd. : Knowledge Engineering and Agent Technology*, IOS Series on Frontiers in AI and Applications, Amsterdam, Netherlands, 2004.
- [164] R. V. O'NEILL, D. DEANGELIS, J. WAIDE et T. ALLEN : *A hierarchical concept of Ecosystems*. Princeton University Press, 1986.
- [165] P.-Y. OUDEYER : *L'auto-organisation de la parole*. Thèse de doctorat, University Paris VI, 2003.
- [166] F. PACHET : Interacting with a musical learning system : the continuator. *In A. S. C. ANAGNOSTOPOULOU, M. Ferrand, éd. : Music and Artificial Intelligence*, vol. 2445 de LNAI, p. 119–132. Springer Verlag, September 2002.

Bibliographie

- [167] P. PAPANETROU, G. KOLLIOS, S. SCLAROFF et D. GUNOPULOS : Discovering frequent arrangements of temporal intervals. *In Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM'05)*, p. 354–361, Los Alamitos, CA, USA, 2005. IEEE Computer Society.
- [168] D. PAULUS et J. HORNEGGER : *Applied Pattern Recognition*. Vieweg, 1998.
- [169] J. PETITOT-COCORDA : *Morphogénèse du sens*, vol. I. PUF, 1985.
- [170] D. PIERSON : *Goals and indications for monitoring*, p. 33–44. New York : McGraw-Hill, Inc, 1998.
- [171] F. PORTET : *Pilotage d'algorithmes pour la reconnaissance en ligne d'arythmies cardiaques*. Thèse de doctorat, l'Université de Rennes 1, 2005.
- [172] F. PORTET, E. REITER, J. HUNTER et S. SRIPADA : Automatic generation of textual summaries from neonatal intensive care data. *In R. BELLAZZI, A. ABU-HANNA et J. HUNTER, eds : Proceedings of the 11th Conference on Artificial Intelligence in Medicine (AIME 07)*, p. 227–236, 2007.
- [173] R. QUINIOU, G. CARRAULT, M.-O. CORDIER et F. WANG : Temporal abstraction and inductive logic programming for arrhythmia recognition from electrocardiograms. *Artificial Intelligence in Medicine*, 28:231–263, 2003.
- [174] E. RAMASSO, M. ROMBAUT et D. PELLERIN : State filtering and change detection using tbm conflict - application to human action recognition in athletics videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(7):944–949, 2007.
- [175] F. RASTIER, M. CAVAZZA et A. ABEILLÈ : *Sémantique pour l'analyse*. Masson, 1994.
- [176] C. RATZÉ, F. GILLET, J.-P. MULLER et K. STOFFEL : Simulation modelling of ecological hierarchies in constructive dynamical systems. *Ecological Complexity*, 4(1-2):13–25, 2007.
- [177] J.-F. ROBINEAU : *Méthode de sélections de variables (parmi un grand nombre) dans un cadre de discrimination*. Thèse de doctorat, Université Joseph Fourier, Grenoble, 2004.
- [178] M. SAEED, C. LIEU, G. RABER et R. MARK : MIMIC II : a massive temporal ICU patient database to support research in intelligent patient monitoring. *In Computers in Cardiology*, p. 641–644, 2002.
- [179] A. SALATIAN et J. HUNTER : Deriving trends in historical and real-time continuously sampled medical data. *Journal of Intelligent Infering Systems*, 13(1-2):47–71, 1999.
- [180] P. SALEMBIER et B. PAVARD : Analyse et modélisation des activités coopératives situées. évolutions d'un questionnement et apports à la conception. *@CTIVITES*, 1(1):87–99, 2004.
- [181] S. SALVADOR et P. CHAN : FastDTW : Toward accurate dynamic time warping in linear time and space. *In Proceedings of Conference on Intelligent Data Analysis*, 2007.
- [182] C. S. SASSOON et G. T. FOSTER : Patient-ventilator asynchrony. *Current Opinion in Critical Care*, 7(1):28–33, 2001.
- [183] F. SCHLEIERMACHER : *Hermeneutics : The Handwritten Manuscripts, traduction J. Duke et J. Forstman*. Missoula : Shcholars, 1977.
- [184] R. SCHOENBERG, D. Z. SANDS et C. SAFRAN : Making ICU alarms meaningful : a comparison of traditional vs. trend-based algorithms. *Proceedings of AMIA Annual Symposium*, p. 379–383, 1999.
- [185] E. SCHWARZ : The Future Evolution of Consciousness as a Dialogue between Individuals and Society. Application of a Holistic Metamodel to the Emergence of a Global Human System. *In J. M. WILBY, éd. : Proceedings of the 40th Annual Meeting of the International Society for the Systems Science, Budapest*. ISSS, 1996.
- [186] Y. SHAHAR : A framework for knowledge-based temporal abstraction. *Artificial Intelligence*, 90(1-2):79–133, 1997.
- [187] Y. SHAHAR, D. BOAZ, G. TAHAN, M. GALPERIN, D. GOREN-BAR, H. KAIZER, L. V. BASSO, S. B. MARTINS et M. K. GOLDSTEIN : Interactive visualization and exploration of time-oriented clinical data using a distributed temporal-abstraction architecture. *Proceedings of AMIA Annual Symposium*, p. 1004, 2003.
- [188] Y. SHAHAR et C. COMBI : Timing is everything. time-oriented clinical information systems. *Western Journal of Medicine*, 168(2):105–113, 1998.
- [189] Y. SHAHAR, D. GOREN-BAR, D. BOAZ et G. TAHAN : Distributed, intelligent, interactive visualization and exploration of time-oriented clinical data and their abstractions. *Artificial Intelligence in Medicine*, 38(2):115–135, 2006.
- [190] Y. SHAHAR et M. A. MUSEN : Knowledge-based temporal abstraction in clinical domains. *Artificial Intelligence in Medicine*, 8(3):267–298, 1996.
- [191] S. SHARSHAR, L. ALLART et M.-C. CHAMBRIN : A new approach to the abstraction of monitoring data in intensive care. *In Proceedings of the 10th conference on Artificial Intelligence Medicine*, p. 13–22, 2005.

- [192] S. SHARSHAR et M.-C. CHAMBRIN : Vers une analyse de la dynamique des paramètres physiologiques en unité de soins intensifs. In *Actes de l'atelier Extraction de motifs temporels pour la détection en ligne de situations critiques, EGC'05*, 2005.
- [193] P. SHYR, G. TECUCI et M. BOICU : Evaluation of mixed-initiative knowledge base development methods and tools. In *Proceedings of IJCAI-2001 Workshop on Empirical Methods in AI*, p. 47–53, Seattle, Washington, August 2001. AAAI Press.
- [194] A.-S. SILVENT : *Extraction de connaissances pour la construction de scénarios médicaux*. Thèse de doctorat, Université Joseph Fourier, 2004.
- [195] A.-S. SILVENT, M. DOJAT et C. GARBAY : Multi-level temporal abstraction for medical scenarios construction. *International Journal on Adaptive Control*, 19(5):377–394, 2005.
- [196] B. W. SILVERMAN : *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, London, 1989.
- [197] C. SINDERBY : Neurally adjusted ventilatory assist (NAVA). *Minerva Anesthesiol*, 68(5):378–380, 2002.
- [198] M. STACEY et C. MCGREGOR : Temporal abstraction in intelligent clinical data analysis : A survey. *Artificial Intelligence in Medicine*, 39(1):1–24, 2007.
- [199] L. STEELS : Perceptually grounded meaning creation. In M. TOKORO, éd. : *Proceedings of the Second International Conference on Multi-Agent Systems, ICMAS-96*, p. 338–344, Menlo Park, CA, 1996. The AAAI Press.
- [200] L. STEELS : The origins of syntax in visually grounded robotic agents. *Artificial Intelligence*, 103(1-2):133–156, 1998.
- [201] L. STEELS : Evolving grounded communication for robots. *Trends in Cognitive Science*, 7(7):308–312, 2003.
- [202] L. STEELS : *Symbols, Embodiment and Meaning*, chap. The symbol grounding problem has been solved. So what's next ? Oxford University Press, 2007.
- [203] L. STEELS et F. KAPLAN : Collective learning and semiotic dynamics. In D. FLOREANO, J.-D. NICLOUD et F. MONDADA, édés : *ECAL99*, p. 679–688. Springer-Verlag, 1999.
- [204] L. STEELS et F. KAPLAN : Bootstrapping grounded word semantics. In T. BRISCOE, éd. : *Linguistic evolution through language acquisition : formal and computational models*, p. 53–74. Cambridge University Press, 2002.
- [205] F. STEIMANN : *Diagnostic Monitoring of Clinical Time Series*. Thèse de doctorat, Technischen Universität Wien, 1965.
- [206] F. STEIMANN : The interpretation of time-varying data with DIAMON-1. *Artificial Intelligence in Medicine*, 8(4):343–358, 1996.
- [207] L. SUCHMAN : *Human-Machine Reconfigurations - Plans and Situated Actions (2nd Edition)*. Cambridge University Press, 2006.
- [208] J. THEUREAU : *Handbook of cognitive task design*, chap. Course-of-action analysis and course-of-action centered design, p. 55–81. 2003.
- [209] A. THILLE, P. RODRIGUEZ, B. CABELLO, F. LELLOUCHE et L. BROCHARD : Patient-ventilator asynchrony during assisted mechanical ventilation. *Intensive Care Medicine*, 32(10):1515–1522, 2006.
- [210] C. L. TSIEN : Reducing false alarm in the intensive care unit : A systematic comparison of four algorithms. In *Proceedings of AMIA Annual Symposium*, Cambridge, Massachusetts, 1997.
- [211] C. L. TSIEN : Event discovery in medical time-series data. In *Proceedings of AMIA Annual Symposium*, p. 858–862, 2000.
- [212] C. L. TSIEN : *TrendFinder : Automated Detection of Alarmable Trends*. Thèse de doctorat, Massachusetts Institute of Technology, 2000.
- [213] C. L. TSIEN et J. C. FACKLER : Poor prognosis for existing monitors in the intensive care unit. *Critical Care Medicine*, 25(4):614–619, 1997.
- [214] C. L. TSIEN, I. S. KOHANE et N. MCINTOSH : Multiple signal integration by decision tree induction to detect artifacts in the neonatal intensive care unit. *Artificial Intelligence in Medicine*, 19(3):189–202, 2000.
- [215] J. UEXKÜLL : *Kompositionslehre der Natur*. Frankfurt am Main, 1980.
- [216] F. VARELA : *Autonomie et connaissance - essai sur le vivant*. Seuil, 1989.
- [217] F. VARELA : *Invitation aux sciences cognitives*. Points, 1997.
- [218] F. VARELA, E. THOMPSON et E. ROSCH : *Inscription corporelle de l'esprit - sciences cognitives et expérience humaine*. Seuil, 1999.

Bibliographie

- [219] M. VERDUIJN, A. DAGLIATI, L. SACCHI, N. PEEK, R. BELLAZZI, E. de JONGE et B. de MOL : Comparison of two temporal abstraction procedures : A case study in prediction from monitoring data. *In Proceedings of AMIA Annual Symposium*, p. 749–53, 2005.
- [220] S. B. WACHTER, B. MARKEWITZ, R. ROSE et D. WESTENSKOW : Evaluation of a pulmonary graphical display in the medical intensive care unit : an observational study. *Journal of Biomedical Informatics*, 38(3):239–243, 2005.
- [221] M. WATTENBERG : Sketching a graph to query a time-series database. *In CHI '01 : CHI '01 extended abstracts on Human factors in computing systems*, p. 381–382, New York, NY, USA, 2001. ACM Press.
- [222] I. WEISS : *L'interprétation*. Ellipses, 2002.
- [223] J. WILLAMOWSKI : *Modélisation de tâches pour la résolution de problèmes en coopération système-utilisateur*. Thèse de doctorat, Université Joseph Fourier - Grenoble 1, 1994.
- [224] P. WILSON : *Computer supported cooperative work : an introduction*. Kluwer Academic, 1991.
- [225] D. WOODS, E. ROTH et K. BENETT : *Cognition, Computing and Cooperation*, chap. Explorations in joint human-machine cognitive systems, p. 123–158. Ablex, 1990.
- [226] E. XING, A. NG, M. JORDAN et S. RUSSELL : Distance metric learning, with application to clustering with side-information, 2003.
- [227] Q. ZHAO et S. S. BHOWMICK : Sequential pattern mining : A survey. Rap. tech., Nanyang Technological University, Singapore, 2003.

Annexes

A Publications

- [1] T. GUYET, C. GARBAY et M. DOJAT, Knowledge construction from time series data using a collaborative exploration approach, *Journal of Biomedical Informatics*, 40 :672-687. 2007.
- [2] T. GUYET, Annotation : A support for co-interpretation, In A. LUCIANI & C. CADOZ édés. : *Proceedings of ENACTIVE 2007 : Fourth International Conference on Enactive Interfaces, Enaction in art*, p. 113-116. 2007.
- [3] T. GUYET, C. GARBAY et M. DOJAT, A Human-Machine cooperative approach for time series data interpretation, In R. BELLAZZI, A. ABU-HANNA & J. HUNTER édés. : *Proceedings of the 11th Conference on Artificial Intelligence In Medicine, AIME'07*, p. 3-12. Springer-Verlag, 2007.
- [4] T. GUYET, C. GARBAY et M. DOJAT, Computer/Human structural coupling for data interpretation, In B. BARDY & D. MOTTET édés. : *Proceedings of ENACTIVE 2006 : Third International Conference on Enactive Interfaces, Enaction and Complexity*, p. 71-72. 2006.
- [5] T. GUYET, C. GARBAY et M. DOJAT, Algorithme d'apprentissage de scénarios à partir de séries symboliques temporelles, In R. QUINIOU & G. HÉBRAIL édés. : *Atelier Fouille de Données Temporelles, EGC'06 : Extraction et Gestion de Connaissances*. 2006.
- [6] T. GUYET, C. GARBAY et M. DOJAT, Human/Computer interaction to learn scenarios from ICU multivariate time series, In S. MIKSCH, J. HUNTER & E.T. KERAVNOU édés. : *Proceedings of the 10th Conference on Artificial Intelligence in Medicine, AIME'05*, p.424-428. Springer-Verlag, 2005.
- [7] T. GUYET, C. GARBAY et M. DOJAT, Partenariat utilisateur-SMA pour l'apprentissage à partir de séries temporelles, application au monitoring en soins intensifs, In A. DROGOUL & E. RAMAT édés. : *Actes des JFSMA'05 : Journée Francophone des Systèmes Multi Agents, Vers la Conception de Système Artificiels Socio-Mimétiques*, p. 15-18. Hermès Science, 2005.
- [8] T. GUYET, C. GARBAY et M. DOJAT, Apprentissage de scénarios à partir de séries temporelles multivariées, In S. PINSON & N. VINCENT édés. : *Actes des Cinquièmes Journées Extraction et Gestion des Connaissances*, p. 133. Cépaduès-Éditions, 2005.

B Algorithme MNClassification

On donne dans cet annexe quelques précisions sur l'algorithme de classification MN qui est utilisé pour classer les représentations vectorielles des segments.

B.1 Algorithme de classification

1. Tirer un exemple e au hasard dans la liste des éléments à classer.
2. Parmi les classes existantes, chercher la classe C qui convient le mieux à e . Au besoin, on crée une nouvelle classe C .
3. Intégrer e à C
4. Adapter C , *i.e.* ajuster la distance aux données.
5. Vérifier la consistance de C
 - Si un élément n'est pas consistant, on le replace dans la liste des éléments à classer et on repasse à 4)
 - Sinon, on passe à 6)
6. Vérification de la typicité des classes
7. S'il reste des éléments dans la liste des éléments, revenir à (1)

B.2 Description des étapes de l'algorithme itératif

On s'attache dans cette section à donner des précisions sur le fonctionnement des différentes étapes du schéma général de la section C.2.3.

B.2.1 Choix de la meilleure classe pour un exemple

Il n'existe pas de prototype de classe. Donc pour pouvoir être intégré à une classe, un exemple doit vérifier la condition de limite de seuil avec l'ensemble des exemples. C'est-à-dire qu'un exemple peut être ajouté à une classe C si on a :

$$\forall e_C \in C, d_C(e_C, e) < s$$

Il faut ensuite, choisir parmi les différentes classes qui respectent cette condition. Cette situation peut arriver au début de la classification où les classes ne sont pas trop proto-"typées", mais ceci doit normalement se faire de plus en plus rare (*i.e.* une seule classe respecte cette condition) au fur et à mesure que les méta-distances inter-classes s'accroissent. Pour cela, on se base sur la distance moyenne sur les exemples d'une classe pour choisir la classe qui correspond le mieux.

Si aucune classe n'a respecté cette condition, une nouvelle classe est créée et elle est alors constituée du seul exemple e .

B.2.2 Vérification de la consistance d'un classe

Dans la mesure où il n'y a pas de point de référence dans une classe, la vérification de la consistance d'une classe consiste à savoir si les distances d'un exemple aux autres exemples sont inférieures à s . Si c'est le cas, il faut alors éjecter un exemple ou les deux.

B.2.3 Éjection d'un exemple

L'éjection d'un exemple consiste à l'enlever de la classe dans laquelle il se situe et de le replacer dans les exemples à classer.

L'éjection d'un exemple est associée à un paramètre r : le seuil d'éjection de la classe. Ce paramètre exprime la contrainte de conservation de consistance d'une classe. Si deux exemples sont éloignés d'une distance supérieure à r , alors l'un des deux n'est plus consistant pour cette classe. Deux approches peuvent être proposées pour résoudre cette situation indésirable :

- on éjecte les deux exemples de la classe,

- on éjecte un seul exemple de la classe.

Les deux approches permettent, *a priori*, à la classe de recouvrer sa consistance. Si deux exemples sont supprimés, on risque d'éjecter un exemple qui est pourtant significatif à la classe et donc ralentir la convergence, puisque cet exemple sera à placer dans cette classe de nouveau. Tandis que si on éjecte uniquement un de ces exemples, on évite ce problème et s'il s'avérait que les deux n'étaient pas viables dans cette classe, le second sera éjecté plus tard dans le processus d'adaptation de la classe. On fait le choix de n'éjecter qu'un seul exemple de la classe. Le problème qui se pose maintenant est de savoir lequel des deux exemples est à éjecter.

L'approche que nous proposons est de ne pas éjecter dès que deux exemples sont séparés d'une distance supérieure à r , mais lorsqu'un exemple vérifie cette condition avec plusieurs autres exemples de la classe. On introduit alors le paramètre tE qui représente le taux d'éjection au delà duquel un exemple e doit être éjecté. Le taux d'éjection est le pourcentage d'exemples avec lequel un exemple e ne vérifie pas la contrainte de distance r . On a préféré un taux qu'un nombre d'exemples pour ne pas faire de distinction de traitement entre des classes contenant des nombres d'exemples différents. Ce paramètre est par défaut fixé à 50% pour indiquer qu'il faut que pour qu'un exemple soit éjecté d'une classe il faut qu'il soit différent de la majorité des exemples, sinon il reste dans la classe.

B.2.4 Adaptation d'une distance

L'adaptation de la distance d'une classe est nécessaire dès qu'un exemple est (i) ajouté ou (ii) supprimé de la classe. L'adaptation est une alternative à l'utilisation d'un algorithme de *feature selection* pour caractériser la classe. Dans cette section, on présente l'algorithme sur des données vectorielles et avec des distances euclidiennes pondérées. L'adaptation de la distance consiste en une adaptation des poids attribués à chaque caractéristique (ou plutôt à une répartition des poids puisque ceux-ci représentent un vecteur α qui est ensuite normalisé). Le mécanisme d'adaptation de la distance est le même pour l'ajout ou la suppression d'un exemple, mais les sens de l'adaptation seront opposés.

Soit α^t (resp. α^{t+1}), le vecteur de pondération de la distance d'une classe avant (resp. après) adaptation. L'adaptation est effectuée de la façon suivante :

$$\alpha_i^{t+1} = \frac{\alpha_i^t \pm \beta}{\|\alpha_i^t \pm \beta\|}.$$

Lorsqu'il s'agit d'un ajout (resp. d'une suppression) à la classe d'un exemple e , on ajoute (resp. soustrait) le vecteur β . Les éléments du vecteur β sont soit eA , soit 0, soit $-eA$, selon qu'il faille renforcer, ne pas modifier ou inhiber une caractéristique de la classe :

- Si $\beta[i]^7$ est positif, alors $\alpha_{t+1}[i]$ aura **plus** d'importance dans α que $\alpha_t[i]$ dans α .
- Si $\beta[i]$ est nul, alors $\alpha_{t+1}[i]$ aura **autant** d'importance dans α que $\alpha_t[i]$ dans α .
- Si $\beta[i]$ est négatif, alors $\alpha_{t+1}[i]$ aura **moins** d'importance dans α que $\alpha_t[i]$ dans α .

Ensuite, le vecteur α est de nouveau normalisé.

La valeur positive eA fixe donc le degré de l'adaptation lors d'un ajout ou retrait d'un exemple. Cette valeur est constante dans tout le processus. Plus cette valeur est petite, plus l'adaptation sera lente. Les vecteurs α étant normalisés, on peut donner une valeur universelle au paramètre eA .

On décrit maintenant comment le vecteur β est construit à partir de e . On va pour cela introduire un nouveau paramètre fS , puis nous donnerons l'algorithme permettant de calculer β .

fS exprime la similarité entre les caractéristiques de deux exemples e_1 et e_2 . On rappelle ici qu'une caractéristique d'un exemple est ici une des valeurs du vecteur sur la i -ème dimension par exemple. Ainsi la similarité entre les i -ème caractéristiques de deux exemples e_1 et e_2 est la distance euclidienne entre la valeur sur la i -ème dimension du vecteur e_1 et celle du vecteur e_2 ⁸. Lorsque la distance⁹ entre deux caractéristiques est inférieure à fS , les deux caractéristiques sont *proches*, sinon elles sont considérées comme *différentes*.

L'algorithme qui permet de calculer β (*i.e.* déterminer si on met $-eA$ ou $+eA$) est le suivant :

⁷On représente ici par $v[i]$, la i -ème composante du vecteur v

⁸On compare toujours des caractéristiques identiques des deux vecteurs (*e.g.* on ne compare pas la 3ème caractéristique de l'un avec la 5ème de l'autre).

⁹Distance euclidienne entre deux nombres.

Pour chaque caractéristique i

Pour chaque exemple e_C de la classe C

si $e_C[i]$ proche de $e[i]$, alors $\beta[i] = \beta[i] + 1$

sinon $\beta[i] = \beta[i] - 1$

fin

fin

$$\beta = eA * \beta / |\beta|$$

B.2.5 Vérification de typicité des classes

Dans cette dernière étape, il s'agit de participer à l'ajustement du nombre des classes à partir du critère de maximisation des méta-distances inter-classes. La création des classes est effectuée lorsqu'un exemple ne trouve de place initialement au début d'un cycle de classification. Les suppressions de classes interviennent en fin de cycle en utilisant deux mécanismes :

- Fusion de classes proches
- Suppression des classes d'"orphelin"

La fusion de classe intervient lorsque deux classes sont trop proches. Dans notre exemple, on utilise une distance euclidienne entre les vecteurs α de deux classes. Si cette distance est inférieure à un seuil fixé fT , alors les classes sont considérées comme proches et sont fusionnées : les exemples sont regroupés dans une seule classe et le vecteur α de la distances caractéristique est construite comme le vecteur moyen des deux vecteurs α normalisés (Celui-ci devra être proche des deux précédents car initialement ils étaient proches).

Lorsqu'une classe ne comporte plus qu'un seul exemple, on parle d'une classe d'orphelin. Cette classe en l'état n'est pas utile et l'exemple qui y est contenu ne pourra être jamais être supprimée par les mécanismes d'éjection proposés. Par conséquent, si on ne veut pas obtenir une classification comportant un nombre croissant de classes orphelines, on supprime celles-ci. L'exemple d'une classe orpheline est alors replacé dans la liste des éléments à classer et la classe est détruite (supprimée de la classification).

Les détails de l'algorithme ont laissé apparaître un grand nombre de paramètres nécessaires. On fait maintenant le bilan sur les différents paramètres à définir pour notre classification en essayant de convaincre le lecteur que cette difficulté peut être dépassée et voire même utilisée pour la classification.

B.3 Paramètres de l'algorithme

Trois paramètres sont nécessaires et spécifiques à l'algorithme, on donne donc quelques détails sur leur signification. Ils doivent être définis en accord avec les données et les résultats attendus. Les autres paramètres sont ensuite listés. Ces derniers sont nombreux, mais on peut constater qu'ils sont, pour la plupart, liés à l'utilisation de la méthode d'adaptation de la distance Euclidienne pondérée du paragraphe précédent. Cette méthode n'est d'une part qu'une possibilité pour implémenter une distance adaptative et d'autre part, à notre avis, les paramètres utilisés peuvent se voir attribuer des valeurs assez universelles.

B.3.1 Le seuil de similarité (s)

Une valeur s indique le seuil de similarité intra-classe des exemples. Pour intégrer un exemple au sein d'une classe C , celui-ci doit satisfaire une intra-distance (distance de la classe) à chaque exemple de C inférieure à s .

B.3.2 Seuil de fusion de classe (fT)

La fusion de classe assure que la distance entre deux classes sera nécessairement supérieure à fT . On a ainsi un paramètre pour fixer la distance inter-classe minimum à respecter.

B.3.3 Le seuil d'éjection d'un exemple d'une classe (r)

Le paramètre r donne le seuil d'éjection d'un exemple de la classe. On distingue cette valeur de s . Si un exemple a une de ses intra-distances supérieure à r , il est "éjecté" de celle-ci et devra être reclassé (En pratique il est replacer dans l'ensemble des éléments à classer).

On a ajouté le paramètre r pour stabiliser du processus, *i.e.* éviter aux exemples de faire des passages trop furtifs dans une classe. De tels comportements limiteraient les successions d'adaptations qui permettent de spécifier une classe, et par conséquent cela ne faciliterait pas la convergence du processus de classification. Pour gagner en stabilité, on doit avoir $r < s$. L'écart entre les deux valeurs définit une "zone franche" dans laquelle les exemples sont en sursis de la classe, *i.e.* à la limite d'être éjecté pour être reclassé. Plus la valeur de r sera proche de s , moins l'algorithme est stable.

B.3.4 Autres paramètres

D'autres paramètres ont également été introduit dans l'algorithme, mais ils n'ont pas autant d'influence sur la classification que les autres et peuvent être fixés *a priori* pour tous les jeux de données.

- tE : Taux d'éjection
- fS : Similarité de caractéristiques
- eA : Adaptation élémentaire

B.4 Intégration d'un utilisateur

Sous cette forme, la classification est très adaptative à des modifications extérieures et est donc idéale pour envisager de la faire fonctionner en partenariat avec un utilisateur. La flexibilité de la classification permet d'intégrer à tout moment de nouveaux exemples. Mais l'algorithme permet également l'utilisation de liens entre exemples à classer qui permet à l'utilisateur d'introduire une partie de sa connaissance sur le jeu d'exemple à classer sans limiter la possibilité de l'algorithme à construire ses propres classes.

B.4.1 Ajout de liens entre exemples

Un utilisateur peut exprimer facilement la similarité entre deux exemples d'un jeu à classer. Un lien entre deux exemples exprime cette similarité entre eux.

L'algorithme reste exactement le même à cela près qu'on force deux exemples liés à appartenir à une même classe. Il y a donc des modifications dans les étapes d'insertion et d'éjection d'un exemple d'une classe. Si un exemple lié à un second est introduit dans une classe ou éjecté dans la liste des exemples à classer, alors le second le suit automatiquement. Pour l'adaptation de la classe, les deux exemples participent à adapter celle-ci. Les conséquences seront donc plus importante mais reflète bien le contenu de la classe, car l'exemple introduit par suivi du lien n'a pas à vérifier initialement le critère de proximité aux autres exemples pour être introduit, mais il a à respecter le critère de consistance.

Le suivi des liens est transitif, c'est-à-dire que si un exemple lié à un premier introduit dans une classe est lui-même lié à un troisième, ce sont les trois qui seront introduit dans la classe.

Potentiellement, dans une chaîne de liens, deux exemples peuvent être très différents au regard de la classification en cours et il peut être dommageable d'intégrer tous ces exemples. En effet, s'ils sont très différents, d'une part l'introduction d'un seul coup de tous ces exemples dans une classe peut provoquer un bouleversement de la distance (vecteur α) de la classe, et d'autre part on augmente la probabilité qu'ils soient tous éjectés de la classe par l'éjection d'un seul de l'un d'eux. Par conséquent, dans l'algorithme, on peut adopter une stratégie d'introduction de liens limités en profondeur (*i.e.* le nombre de liens entre exemples et l'exemple introduit initialement. Dans l'exemple précédent le troisième exemple était à la profondeur 2 puisqu'il y avait deux liens entre l'exemple et l'exemple introduit).

B.4.2 Introduction de classes et exemples fixes

En plus de liens, un utilisateur peut être intéressé par initier manuellement la construction de la classification en créant des classes qui ne pourront être détruites (on les dit alors "fixées") mais dont la distance sera quand même adaptée par le processus. L'intérêt de telles classes est d'y introduire des

exemples également fixées (ils ne pourront pas être déplacés de la classe) qui permettent d'initier une caractérisation de cette classe. On permet ainsi de superviser partiellement la classification.

Le reste des exemples du jeu de données se greffe autour de ces classes déjà existantes. En suivant le même processus le processus permet de créer éventuellement d'autres classes et d'intégrer des exemples aux classes fixées.

B.5 Conclusions et perspectives

Il a été noté que l'algorithme de classification est central dans la dynamique globale du système puisque c'est de lui que dépend la construction des événements. Il répond aux attentes fortes que nous avons fixées pour la classification de segments et la construction de modèle d'événements :

- les résultats de classification sont corrects. Pour nos données d'application, ils sont pas très inférieurs aux résultats d'un l'algorithme des K-Means,
- la méthode permet de prendre en compte des liens entre les exemples. Les expérimentations sur l'introduction de liens ne sont pas très significatifs, mais suffisamment encourageant pour explorer plus à fond cet algorithme.

Le paramétrage de l'algorithme reste un point sensible de l'algorithme. Pour les deux paramètres principaux s et fT , une connaissance approximative des données (estimation d'une distance entre deux exemples similaires et d'une distance minimum entre deux modèles différents) permet de les fixer relativement intuitivement. De plus, on pourrait se servir des interactions de l'utilisateur pour ajuster ces valeurs individuellement pour chaque classe :

- Les exemples liés d'une classe peuvent servir de référence pour calculer une distance minimum entre deux exemples similaires d'une classe et ainsi ajuster le paramètre s .
- L'utilisation d'*anti-liens* peuvent servir à ajuster la distance minimum à avoir entre deux classes, *i.e.* le paramètre fT . Des *anti-liens* sont des liens indiquant que deux exemples ne sont pas identiques et qu'ils ne peuvent alors jamais appartenir à une même classe.

Pour les paramètres plus avancés qui peuvent prendre également beaucoup d'importance, il est nécessaire d'avoir une bonne compréhension des mécanismes de l'algorithme pour pouvoir les fixer et nous n'avons actuellement pas de solution pour en simplifier la configuration.

Si dans des applications on cherche les performances de classification d'un algorithme, MN n'a peut être pas sa place, mais dans le cadre de la conception de systèmes adaptatifs, il peut trouver une place importante. Par conséquent, il semble intéressant de poursuivre son étude à la fois en réalisant plus d'expérimentations avec des données de classification plus standardisées qui permettraient de se confronter aux résultats d'algorithmes de classification autres et ainsi d'acquérir une meilleur connaissance de ses capacités.

Il nous semble également intéressant de faire une étude théorique d'une classification dans laquelle les distances sont propres à chacune des classes. À notre connaissance, il n'existe pas d'algorithme utilisant une telle approche. L'étude de la topologie de l'espace de représentation de la classification peut permettre de proposer un nouvel algorithme sur cette approche qui aura des résultats plus en accord avec les attentes de la problématiques de classification que l'algorithme MN (en particulier être capable de fournir toujours le même résultat de classification). De plus, une formalisation mathématique de notre algorithme permettrait peut être d'apporter une preuve de convergence même si elle nous conduit à modifier certains éléments (*e.g.* le mode d'adaptation des distances).

C Algorithme d'apprentissage de motifs temporels

C.1 Algorithme pour la prise en compte de l'information temporelle

On présente ici un algorithme qui a été proposé dans le cadre de cette thèse [105] pour permettre un apprentissage de scénarios à partir de séries symboliques temporelles et découvrant aussi bien la structure symbolique des scénarios que les relations temporelles qui les spécifient. On introduit une modèle de séquence sous la forme d'hypercubes, ensuite, nous décrivons le principe de cet algorithme et enfin nous donnons le pseudo code et nous calculons finalement sa complexité.

C.1.1 Modèle de motif en hyper-cubes

Les exemples dont on se sert pour notre algorithme sont des motifs. À chaque symbole, y compris s'il y a plusieurs occurrences de symboles de même nature dans un motif, est associé une dimension (échelle des temps) sur laquelle on délimite la position dans le temps du symbole. Un motif comportant N symboles, dit de taille N , sera représenté dans un espace à N dimensions. En intersectant l'ensemble des contraintes quantitatives, on définit ainsi un hypercube unique pour chaque motif. Cet hypercube est en effet la représentation géométrique du motif. Dans [61], on pourra trouver des détails sur ce type de représentation qui est appliquée, plus généralement, à des données de type intervalle. On pourra trouver des généralisations des méthodes statistiques à ce type de données chez [79]. D'autres représentations géométriques de motifs sont discutées dans [132]. Si dans la suite on ne s'intéresse qu'aux contraintes quantitatives, on peut donner des interprétations de relations qualitatives temporelles à l'aide de la position des sommets par rapport aux hyperplans diagonaux.

Afin de mieux faire comprendre ces idées, on présente cette représentation pour les motifs de tailles 3. Le motif comportant 3 symboles A , B et C tel que le montre la figure 3 (à gauche) peut être représenté dans un espace à 3 dimensions par le cube (un 3-hyperrectangle) de la figure 3 (à droite). Chaque dimension représente le temps pour un évènement.

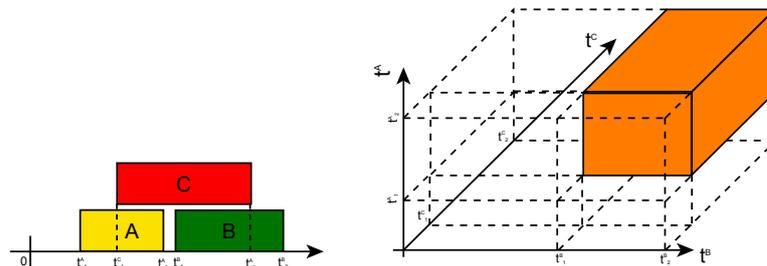


FIG. 3. Motif de taille 3 : à gauche le motif, et à droite sa représentation en 3 dimensions.

C.1.2 Algorithme

L'algorithme construit à partir des motifs exemples par agrégation progressive de symboles le plus grand motif commun à ces exemples et de fréquence supérieure à f (f est un paramètre de l'algorithme) reprenant ainsi l'idée de l'algorithme **A Priori**. Mais dans notre représentation, l'agrégation d'un nouveau symbole signifie augmenter d'une dimension les motifs en cours de construction.

L'algorithme, dont le pseudo-code est donné en annexe, est la répétition de successions de deux étapes, à savoir la sélection des motifs de dimension N suffisamment fréquents, et la construction des candidats-motifs de dimension $N + 1$ à partir de ceux de dimension N , qui sont présentés en détails dans les paragraphes suivants. L'arrêt se fait lorsque qu'il n'y a plus de candidat possible à construire ou que ceux-ci ne sont plus suffisamment fréquents.

Sélection des motifs de dimension N communs aux exemples Soit k le nombre de symboles différents constituant les séries symboliques. Dans le cas le plus général, il y a au plus $k * N$ types de motifs de dimension N possibles, par la suite on limitera ce nombre. On associe à chacun un hyper-histogramme : histogramme en dimension $N + 1$. Pour chaque exemple, on extrait les motifs de dimension N et on empile

l'($N + 1$)-hypercube (de base l'hyperrectangle défini par le motif et de hauteur 1) sur l'hyperhistogramme correspondant (*i.e.* celui où la base de l'hyperhistogramme est l'espace de représentation du motif). La figure 4 donne l'exemple de l'empilement d'hypercube de dimension 2, pour former un histogramme en dimension 3 (c'est le seul cas visualisable).

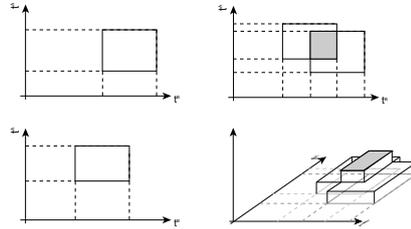


FIG. 4.
Empilement de motifs de taille 2.

Cet histogramme peut être vu comme une approximation (modulo la loi des grands nombres) de la probabilité du motif de dimension N . L'utilisation d'un algorithme de recherche de mixture de gaussiennes (Algorithme EM ou dérivé pour lequel on prend des exemples tirés au hasard avec des probabilités définies par l'historgramme) permet alors de définir ce motif comme une gaussienne à $2N$ paramètres : N moyennes et N variances pour la date moyenne et la durée moyenne de chaque symbole dans le motif. Si on permettait d'avoir des covariances non nulle, pouvant être traduit par l'interdépendance des durées moyennes des motifs, alors il faudrait N paramètres de moyenne et une matrice de covariance de dimension N : pour simplifier dans cette première version de l'algorithme les calculs et l'explication, on ne considère pas ces co-variances. Dans [61], d'autres méthodes d'analyses factorielles permettent de faire cette même étape.

Passage à la dimension supérieure On cherche maintenant à construire des candidats-motifs de dimension $N + 1$ à partir des motifs de dimension N . L'idée de l'algorithme **A Priori** qui est reprise ici peut être traduite par un théorème dont la preuve est triviale par l'absurde.

Théorème : Un motif de dimension $N + 1$ ne peut être de fréquence supérieure à f si tous les sous-motifs qui le composent (en particulier ceux de dimension N) ne sont pas au moins de fréquence f .

On va donc construire les candidats à partir des motifs de dimension N , ils sont moins nombreux et on peut ainsi réduire l'espace de recherche au fur et à mesure que la dimension augmente. De plus, on sait que *a posteriori*, les motifs de dimension N correspondent à des projections suivant la dimension complémentaire associée à l'un des symboles possibles du motif. La figure 5 illustre cette construction.

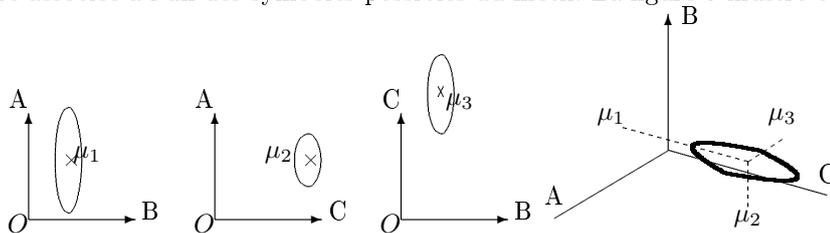


FIG. 5.
Illustration de la construction de motifs de taille 3 à partir de motifs de tailles 2.

Par conséquent, deux conditions sont nécessairement à vérifier par les candidats-motifs. Si le motif est constitué des symboles $(S_1, S_2, \dots, S_{N+1})$, il faut :

1. (Condition du théorème) $\forall n \in [1, N + 1]$, il existe un motif de dimension N de fréquence supérieure à f avec les symboles :

$$(S_1, S_2, \dots, S_{n-1}, S_{n+1}, \dots, S_{N+1}).$$

2. Les $N + 1$ motifs de dimension N doivent être consistants, *i.e.* qu'ils ne doivent pas violer la condition *a posteriori* de projection.

En pratique, on peut vérifier ces conditions (sur la figure 5) en calculant la distance entre μ_3 et la projection du point en 3 dimensions construit à partir de μ_1 et μ_2 sur le plan BC . Une fois que ces conditions ont été vérifiées, on passe à une nouvelle étape de sélection des motifs de taille $N + 1$ maintenant. On considère ainsi des motifs de taille de plus en plus grandes élaguant ceux qui ne nous conviennent pas.

C.2 Pseudo-code de l'algorithme

```

procédure A Priori ( $f$  : fréquence min,  $E$  : les exemples)
début
   $L_2 \leftarrow \{\text{Motifs de dimension 2, fréquents}\}$ 
   $k \leftarrow 2$ 
  Tant que  $L_k \neq \emptyset$  faire
    Pour tout  $m \in L_k$  faire
       $h \leftarrow \text{CreerHistogramme}(m, E)$ 
       $M_m \leftarrow \text{CreerMotifs}(h, f)$ 
    Fin pour tout
     $L_{k+1} \leftarrow \text{GenererCandidats}(M)$ 
     $k \leftarrow k + 1$ 
  Fin tant que
fin
procédure CreerHistogramme ( $m$  : Espace support,  $E$  : Series Exemples)
début
   $H \leftarrow \text{Histogramme plat de base } m$ 
  Pour tout  $ex \in E$  faire
     $H(ex) \leftarrow H(ex) + 1$ 
  Fin pour tout
  retourner  $H$ 
fin
procédure CreerMotifs ( $H$  : Histogramme,  $f$  : Fréquence minimum)
début
   $M \leftarrow \text{Résultat de l'algorithme EM}$ 
   $M \leftarrow \text{Suppression de } M \text{ des Gaussiennes trop 'faibles' pour } f$ 
  retourner  $M$ 
fin
procédure GenererCandidats ( $M$  : Liste de motifs de dimension  $k$ )
début
   $C \leftarrow \text{Liste vide de motifs de dimension } k + 1$ 
   $C \leftarrow \text{Ensemble des motifs de dimension } k + 1 \text{ vérifiant les 2 contraintes : A Priori et A Posteriori}$ 
  retourner  $C$ 
fin

```

C.3 Estimation de la complexité

Soient M le nombre de symboles, P le nombre d'exemples et N le taille des motifs finaux. En supposant que jusqu'au dernier rang toutes les combinaisons de symboles restent suffisamment fréquentes et simples (pas de motifs différenciés par les contraintes temporelles uniquement), alors le calcul donne la complexité de l'algorithme de l'ordre de :

$$\frac{1 - M^{N+1}}{1 - M} (\text{CoutEM} + P + M)$$

Comme dans l'algorithme **A Priori**, le coût est exponentiel en fonction de la taille du motif appris, sans même se poser la question du coût de l'algorithme EM qui augmente exponentiellement également car on doit augmenter exponentiellement le nombre d'exemples pour limiter le fléau de la dimension.

La complexité de l'algorithme ainsi que malédiction de la dimension limiteront la taille des motifs apprenables. Seule l'expérimentation permettra d'estimer la taille des motifs qui pourront être appris, en testant différentes heuristiques.

Table des matières

Introduction	3
1 Amélioration du monitoring en unité de soins intensifs	7
A Monitoring de patients en Unités de Soins Intensifs (USI)	8
A.1 Principes du monitoring d'un système dynamique	8
A.1.1 Des systèmes automatiques de monitoring pour aider les opérateurs	9
A.1.2 Tâches réalisées par un système de monitoring	10
A.2 Le monitoring de patients en USI	14
A.2.1 Unités de soins intensifs	14
A.2.2 Équipement d'une salle d'USI	15
B Amélioration des systèmes de monitoring de patients en USI	16
B.1 Amélioration des performances du monitoring	16
B.1.1 Amélioration de la chaîne de traitement de l'information	16
B.1.2 Alarmes plus informatives	17
B.1.3 Déclencher au plus tôt les alarmes	17
B.2 Amélioration des services d'un système de monitoring	17
B.2.1 Aide à la configuration ou à la re-configuration	18
B.2.2 Suivi automatique de protocole	18
B.3 Voies d'amélioration des systèmes de monitoring	18
B.3.1 Systèmes actuels	19
B.3.2 Extraction de tendances	19
B.3.3 Utilisation de modèles complexes d'évènements	20
B.3.4 Fusion multi-sources des informations	21
B.3.5 Reconnaissance de scénarios	21
B.3.6 Configuration automatique du monitoring	22
C Construction de modèles pour le monitoring	22
C.1 Formalismes pour les modèles d'évènements et de scénarios	23
C.2 Problématique de la construction des modèles	24
D Des systèmes pour aider à l'exploration de séries temporelles	25
D.1 Systèmes de parcours de séries temporelles	25
D.1.1 Interroger les séries temporelles intuitivement	25
D.1.2 Représenter l'information autrement	26
D.2 Système d'aide à l'exploration de séries temporelles	29
D.2.1 TimeSearcher 2 et 3	30
D.2.2 KNAVE-II	31
D.3 Systèmes pour l'annotation de séries temporelles médicales	32
D.4 Positionnement de notre proposition	33
2 Interprétation collaborative de séries temporelles	35
A Interprétation	36
A.1 Caractéristiques d'une tâche d'interprétation	36
A.1.1 Cercle herméneutique	37
A.1.2 Appropriation et contextualisation de l'interprétation	38
A.1.3 Objectif inatteignable de cohérence	39
A.2 Interprétation et émergence de "sens"	40
A.2.1 Émergence	40
A.2.2 Émergence de "sens"	44
A.2.3 Exemples d'opérationnalisations de l'émergence de sens	46

Table des matières

B	Interprétation des séries temporelles	47
B.1	Éléments d'une interprétation de séries temporelles	47
B.1.1	Description des <i>atomes</i> d'une interprétation de séries temporelles	47
B.1.2	Sens porté par les éléments de l'interprétation	48
B.1.3	Abstraction des données	48
B.2	Interprétation de séries temporelles	49
B.2.1	Interprétabilité des traces de systèmes dynamiques	49
B.2.2	Tâche d'interprétation de traces de systèmes dynamiques	49
B.2.3	Opérationnalisation de l'interprétation de séries temporelles	50
C	Collaboration Homme/Machine	52
C.1	Collaboration et Collaboration H/M	52
C.1.1	Caractéristiques d'une machine pour une collaboration Homme/Machine	53
C.1.2	Méthodes d'échange de connaissances pour la compréhension mutuelle	55
C.2	Couplage Homme/Machine : nouveau paradigme de l'Enaction	57
C.2.1	Couplage structurel et Enaction	57
C.2.2	Enaction et informatique	60
C.2.3	Une nouvelle approche de la collaboration Homme/Machine	61
C.2.4	Annotations comme échange de connaissances dans les systèmes couplés	62
D	Interprétation collaborative	63
D.1	Interprétation collaborative à la lueur de l'Enaction	63
D.1.1	Point de vue Enactif de la tâche d'interprétation	63
D.1.2	Annotations et interprétation	64
D.1.3	Construction d'une cohérence entre le système et l'humain	65
D.1.4	Annotation comme moyen d'échange dans la collaboration	65
D.2	Caractéristiques du système pour l'interprétation collaborative	65
D.2.1	Limiter l'introduction de connaissances <i>a priori</i>	66
D.2.2	Adapter la "perception" et la "cognition" de la machine	67
D.3	Collaboration dans l'interprétation de séries temporelles	67
D.3.1	Sémantique annotationnelle	67
D.3.2	Schéma de l'interprétation collaborative	68
3	Un système de support à l'interprétation collaborative de séries temporelles	71
A	Architecture générale : Un système multi-agents	72
A.1	Description du SMA	72
A.1.1	Agentification du processus d'interprétation	73
A.1.2	Organisation : Hiérarchie de triades	73
A.1.3	Définition des agents	75
A.1.4	Définition d'un environnement	77
A.1.5	"Interactions" : Flux d'informations entre les agents	77
A.2	Dynamique des agents	78
A.2.1	Schéma de fonctionnement d'un agent	78
A.2.2	Dynamique de création et suppression des agents	79
A.3	Interactions avec l'utilisateur	80
A.3.1	Utilisateur situé dans la hiérarchie	80
A.3.2	Interactions par les annotations	81
A.3.3	Modification des modèles par les exemples	82
B	Modèles	83
B.1	Choix des modèles et des distances	84
B.1.1	Relation des modèles et distances à la perception	84
B.1.2	Introduction d' <i>a priori</i> par les modèles	84
B.1.3	L'importance des distances	84
B.2	Modèle de segments	85
B.2.1	Distances usuelles entre segments	85
B.2.2	Choix <i>a priori</i> d'une distance	87
B.2.3	Proposition d'un modèle de segment	88
B.3	Modèle d'évènement	89
B.4	Scénarios et explications	90
B.4.1	Modèle de scénarios	90
B.4.2	Explication	91
C	Principaux algorithmes de traitements des données	92
C.1	Segmentation des séries temporelles	92

C.1.1	Méthodes de segmentation de séries temporelles	92
C.1.2	Segmentation par reconnaissance de modèles	94
C.2	Construction de modèles d'évènements	96
C.2.1	Vue générale de la construction de modèles d'évènements	96
C.2.2	Introduction à l'algorithme de classification	96
C.2.3	Algorithme de clustering/classification	98
C.2.4	Classification comme système autopoïétique, adaptatif	100
C.3	Apprentissage de scénarios	101
C.3.1	Construction d'exemples pour une explication	101
C.3.2	Choix d'une méthode parmi les méthodes existantes	102
C.4	Reconnaissance de scénarios	105
C.5	Motivation des choix des algorithmes	106
4	Évaluations	107
A	Méthodologie d'évaluation de l'interprétation collaborative	108
A.1	Cadre méthodologique des évaluations	108
A.1.1	Faisabilité de l'approche Enactive	108
A.1.2	Pertinence de l'approche Enactive	110
A.1.3	Utilisabilité du couplage	110
A.2	Prototype du système pour l'évaluation : CATS	110
A.3	Données	111
A.3.1	Données réelles d'asynchronies Patient/Ventilateur de patients en USI	112
A.3.2	Données simulées	114
B	Évaluations de la faisabilité de la collaboration	117
B.1	Évaluation des algorithmes	117
B.1.1	Algorithme de <i>pattern matching</i> pour la segmentation de séries temporelles	117
B.1.2	Algorithme de classification	120
B.1.3	Algorithme d'Apprentissage et Reconnaissance de scénarios	124
B.2	Expérimentations de la cohérence des interprétations construites	130
B.2.1	Expérimentation de la convergence sur données simulées	130
B.2.2	Résultats	130
B.2.3	Apport de la reconnaissance de scénarios à la convergence de l'interprétation	134
C	Évaluation de la pertinence de la collaboration	139
C.1	Expérimentations	139
C.2	Résultats	140
C.2.1	Évaluations quantitatives	140
C.2.2	Réponses au questionnaire	141
D	Utilisation pour l'interprétation de séries temporelles de patients ventilés	145
5	Discussions et perspectives	151
A	L'Enaction et l'annotation comme approche de l'interprétation collaborative	152
A.1	Introduire de l'intention dans le système	152
A.2	La recherche d'un consensus dans l'interprétation	153
A.3	Schémas Sensori-moteur : Faire des essais/erreurs	154
A.4	Choix des annotations comme moyen de communication	154
B	Réalisation d'un système sur les principes de l'Enaction	155
B.1	Expressivité de l'interprétation	155
B.1.1	Ajouter la dimension contextuelle	156
B.1.2	Problème de la granularité temporelle	156
B.1.3	Manque de focalisation dans la construction des explications	157
B.2	Complexité des algorithmiques	157
B.2.1	Estimation de la complexité des algorithmes	157
B.2.2	Problème pour la mise en place de schémas sensori-moteur	158
B.2.3	Perspective d'amélioration des temps de calcul	159
B.3	Difficulté des traitements des triades intermédiaires	159
C	Réponse de l'approche au problème applicatif	159
C.1	Difficulté de l'application à des données biomédicales	160
C.1.1	Difficultés rencontrées avec nos données	160
C.1.2	Données médicales d'application du système	161
C.2	Nouvelle perspective du système : Aide à l'annotation	162

Table des matières

Conclusion	163
Bibliographie	164
Annexes	174
A Publications	177
B Algorithme MNClassification	178
B.1 Algorithme de classification	178
B.2 Description des étapes de l'algorithme itératif	178
B.2.1 Choix de la meilleure classe pour un exemple	178
B.2.2 Vérification de la consistance d'une classe	178
B.2.3 Éjection d'un exemple	178
B.2.4 Adaptation d'une distance	179
B.2.5 Vérification de typicité des classes	180
B.3 Paramètres de l'algorithme	180
B.3.1 Le seuil de similarité (s)	180
B.3.2 Seuil de fusion de classe (fT)	180
B.3.3 Le seuil d'éjection d'un exemple d'une classe (r)	181
B.3.4 Autres paramètres	181
B.4 Intégration d'un utilisateur	181
B.4.1 Ajout de liens entre exemples	181
B.4.2 Introduction de classes et exemples fixes	181
B.5 Conclusions et perspectives	182
C Algorithme d'apprentissage de motifs temporels	183
C.1 Algorithme pour la prise en compte de l'information temporelle	183
C.1.1 Modèle de motif en hyper-cubes	183
C.1.2 Algorithme	183
C.2 Pseudo-code de l'algorithme	185
C.3 Estimation de la complexité	185
Table des matières détaillée	187
Table des figures et Liste des tableaux	190

Table des figures

1	Schéma général de l'interprétation des séries temporelles.	4
1.1	Évolution de l'état d'un système dynamique.	8
1.2	Chaîne de traitements d'un système de monitoring.	10
1.3	Exemple de Moniteur Cardiaque.	12
1.4	Illustration de calcul de tendances par la méthode CUSUM	20
1.5	Exemples de tendances floues	21
1.6	Agencement des Formalismes, Modèles, Observations et Données	23
1.7	Architecture du système CALICOT.	24
1.8	Interface de QuerySketch.	26
1.9	Interface de TimeSearcher.	26
1.10	Interface de VizTree	27
1.11	Interface de TSEplorer	28
1.12	Résumé textuel d'une série par la méthode BT-45.	29
1.13	Interface de TimeSearcher2	30
1.14	Interface de TimeSearcher3	30
1.15	Interface de KNAVE-II.	31
1.16	Poste d'annotation de MIMIC-II.	33
2.1	Illustration d'un cercle herméneutique.	37
2.2	Organisation hiérarchie en holon.	41
2.3	Triade de Schwarz.	44
2.4	Schéma général d'une interprétation de séries temporelles.	50
2.5	Triade.	51
2.6	Opérationnalisation de l'interprétation de séries temporelles.	51
2.7	Problème de la compréhension dans la collaboration.	54
2.8	Choix d'un niveau de communication entre la machine et l'utilisateur.	55
2.9	Principe de notre collaboration entre la machine et l'humain.	61
2.10	Illustration de la transmission des connaissances par les annotation et les cohérences.	66
2.11	Construction d'une interprétation collaborative des séries temporelles.	69
3.1	Agentification du système.	74
3.2	Architecture générale du système.	75
3.3	Organisation hiérarchique du système (Vision en arbre).	75
3.4	Illustration du fonctionnement d'un agent.	79
3.5	Chronogramme d'exemple de fonctionnement.	79
3.6	Architecture du système incluant l'utilisateur.	81
3.7	Outils qui peuvent être utilisés pour réaliser l'annotation.	82
3.8	Illustration de la difficulté de comparer des segments.	85
3.9	Exemples de mesure de distance Euclidienne entre deux segments.	86
3.10	Distance DTW.	86
3.11	Exemples de mesure de distance DTW entre deux segments.	87
3.12	Comparaison des distances euclidienne, DTW et LCSS.	87
3.13	Formalisme de scénario.	90
3.14	Représentation d'un scénario sous la forme d'une explication	91
3.15	Similarité entre scénarios.	92
3.16	Méthode de segmentation des séries temporelles.	95
3.17	Illustration du masquage d'une zone pour la segmentation.	96
3.18	Classification de segments.	97

Table des figures

3.19	Illustration des éléments de la classification.	99
3.20	Exemple de construction de motifs fréquents.	103
3.21	Algorithme APriori.	104
4.1	Interface du système opérationnel.	111
4.2	Illustration des différentes annotations observables sur une série temporelle univariée.	112
4.3	Illustration des asynchronies patient/ventilateur.	113
4.4	Exemple de signaux de débit et pression respiratoires.	114
4.5	Exemple du résultat d'une simulation de scènes.	115
4.6	Exemple du résultat d'une simulation de séries temporelles.	116
4.7	Évaluation du <i>pattern matching</i> : Courbe ROC.	119
4.8	Évaluation de la classification : Influences des liens.	123
4.9	Évaluation de l'apprentissage : Erreur structurelle	126
4.10	Évaluation de l'apprentissage : Variabilité temporelle	127
4.11	Évaluation de l'apprentissage : Génération de modification	128
4.12	Évaluation de l'apprentissage : Améliorations par la reconnaissance de scénarios	129
4.13	Évolution de l'annotation d'une série temporelle : Étape 1.	131
4.14	Évolution de l'annotation d'une série temporelle : Étape 2.	131
4.15	Évolution de l'annotation d'une série temporelle : Étapes 3 et 4.	132
4.16	Évolution de l'annotation d'une série temporelle : Étape 5.	132
4.17	Évolution de l'annotation d'une série temporelle : Étapes 6 à 8.	133
4.18	Évolution de la similarité entre interprétations successives.	134
4.19	Évolution de l'annotation d'une série temporelle : Étapes 9 à 14.	135
4.20	Évolutions de la similarité dans une interprétation pas à pas : Jeu 1.	136
4.21	Évolutions de la similarité dans une interprétation pas à pas : Jeu 2.	136
4.22	Évolutions de la similarité dans une interprétation pas à pas : Jeu 3.	136
4.23	Évolutions de la similarité dans une interprétation pas à pas : Jeu 4.	136
4.24	Annotation initiale et finale du jeu 1.	137
4.25	Annotation initiale et finale du jeu 2.	137
4.26	Annotation initiale et finale du jeu 3.	137
4.27	Évolutions de la similarité dans une interprétation pas à pas avec scénarios : Jeu 3 et 4.	138
4.28	Annotation finale du jeu 4 avec scènes.	138
4.29	Annotations des débits et pressions respiratoires.	146
4.30	Modèles des cycles du débit respiratoire.	146
4.31	Modèles des cycles de la pression respiratoire.	146
4.32	Annotations des débits et pressions respiratoires avec scènes.	147
4.33	Annotations suivantes des débits et pressions respiratoires.	148
4.34	Modèles des cycles de la pression respiratoire (2).	148
4.35	Annotation alternative des débits et pressions respiratoires.	150
4.36	Résultat de l'annotation	150
5.1	Architecture récursive de la construction d'une interprétation.	156
5.2	Exemple de données filtrées provenant d'un patient monitorés pendant 4 heures.	160
3	Motif de taille 3 : à gauche le motif, et à droite sa représentation en 3 dimensions.	183
4	Empilement de motifs de taille 2.	184
5	Illustration de la construction de motifs de taille 3 à partir de motifs de tailles 2.	184

Liste des tableaux

2.1	Mise en correspondance de l'interprétation et des concepts de l'Enaction	64
3.1	Flux d'informations entre les agents.	78
3.2	Méthodes de segmentation	94
4.1	Évaluation du <i>pattern matching</i> : <i>Spécificité</i>	118
4.2	Évaluation du <i>pattern matching</i> : <i>Sensitivité</i>	118
4.3	Évaluation de la classification : Comparaison des rappels et précisions de MN et KM.	121
4.4	Évaluation de la classification : Matrices de confusion	122
4.5	Évaluation de la classification : Influences des liens (Rappel).	123
4.6	Évaluation de la classification : Influences des liens (Précision).	123
4.7	Évaluation de la classification : Influences des liens (Nb Classes).	124
4.8	Similarité entre l'interprétation des utilisateurs et l'annotation attendue.	140
4.9	Évaluations quantitatives de l'activité des utilisateurs devant le système.	142

Interprétation collaborative de séries temporelles multivariées. Application à des données de réanimation médicale.

Résumé

Cette thèse propose une approche de la collaboration homme-machine, inspirée de la théorie de l'Enaction de Varela, dans le domaine de l'interprétation de données complexes. Cette approche se fonde sur les principes d'autonomie et d'adaptation des partenaires.

L'autonomie des deux partenaires vise à leur permettre de réaliser une interprétation en s'appuyant sur leurs compétences propres qui sont également leurs spécificités enrichissant l'interprétation. Pour préserver leur autonomie, la communication du système avec son partenaire est réalisée au travers d'un environnement commun et au moyen d'annotations porteuses du sens de l'interprétation construite. Ces annotations agissent sur le partenaire comme des perturbations qui le poussent à s'adapter pour mettre en cohérence ses connaissances propres et celles disposées dans l'environnement.

Cette approche est appliquée à la conception d'un système multi-agents pour l'interprétation collaborative de signaux physiologiques enregistrés sur des patients en réanimation médicale, *i.e.* de séries temporelles multivariées. Selon le principe d'autonomie, ce système est capable de construire une interprétation des séries temporelles par la construction (1) de modèles d'évènements et de scénarios (ensembles d'évènements reliés par des relations temporelles) et (2) d'annotations à partir de ces modèles. Selon le principe d'adaptation, les traitements du système et les modèles qu'il construit évoluent pour prendre en compte les annotations du partenaire humain au cours de l'interprétation.

Au travers de l'interprétation réalisée collaborativement, les cliniciens peuvent mieux formaliser leurs connaissances et les transmettre à la machine, guidant ainsi la construction de modèles et permettant à terme l'amélioration des systèmes de monitoring.

Mots clés

Collaboration, Autonomie, Adaptation, Interprétation, Enaction, Annotation, Séries temporelles, Scénarios, Unités de soins intensifs, Asynchronies Patient/Ventilateur, Système Multi-Agents.

Abstract

This thesis proposes an approach of the man/machine collaboration inspired by the Varela's theory of Enaction, in interpretation of complex data. This approach is based on the principles of autonomy and adaptation of partners.

These principles aims to enable them to make an interpretation based on their own skills, which are also their specific enriching the interpretation. To preserve their autonomy, the communication between partners is achieved through a common environment and using annotations that contains the meaning of the interpretation. These annotations disturb the partner and leading it/him to adapt its/his own knowledge to ensure the consistency with annotations in the environment.

This approach is applied to the design of a multi-agent system for the collaborative interpretation of physiological signals recorded on patients in intensive care medical, *i.e.* multivariate time series. According to the autonomy principle, the system is able to construct an interpretation of the time series by the construction (1) of models of events and scenarios (sets of events linked by temporal relations) and (2) of the annotations from these models. According to the principle of adaptation, the treatments and models of the system evolve to take into account the human's annotations in the course of interpretation.

Clinicians can better formalize their knowledge through the collaborative interpretation and transmit them to the machine. Thus, he leads the construction of models and eventually enables the improvement of monitoring systems.

Keywords

Collaboration, Autonomy, Adaptation, Interpretation, Enaction, Annotation, Time Series, Scenarios, Intensive Care Unit, Patient-Ventilator Asynchronies, Multi-Agent System.