



الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

جامعة باجي مختار عنابة

UNIVERSITE BADJI MOKHTAR ANNABA

FACULTE DES SCIENCES DE L'INGENIEUR

DEPARTEMENT D'INFORMATIQUE

Thèse En vue de l'obtention du Diplôme de DOCTORAT en Science

Intitulée:

Contrôle Pédagogique Dans Les Environnements Informatiques d'Apprentissage Humain

Présentée par : M, Mohamed BESNACI

Membres de Jury :

Tahar KIMOUR	Professeur	Président	Université de Annaba
Tahar BENSEBAA	Professeur	Rapporteur	Université de Annaba
Nathalie GUIN	M.C. (HDR)	Co-Rapporteur	Université de Lyon
Mahmoud BOUFAIDA	Professeur	Examineur	Université de Constantine
Yacine LAFIFI	Professeur	Examineur	Université de Guelma

Année universitaire : 2018/2019

THÈSE

CONTRÔLE PÉDAGOGIQUE DANS LES
ENVIRONNEMENTS INFORMATIQUES
D'APPRENTISSAGE HUMAIN

Mohamed BESNACI

Avant-propos

Remerciements

Je tiens à remercier, tout d'abord, mes directeurs de thèse :

Madamme Nathalie GUIN, Maitre de conférences à l'Université Claude Bernard Lyon1, qui m'a énormément aidé et conseillé tout au long de ce travail et qui m'a proposé un cadre de travail très favorable. Je la remercie d'avoir consacré beaucoup de son temps pour les nombreuses relectures de mes documents et articles. Elle a été toujours disponible pour répondre aux questions que je lui posais. Ses remarques m'ont permis de faire progresser ce travail.

Monsieur Tahar BENSEBAA, Professeur à l'Université Badji-Mokhtar Annaba, pour l'encadrement de mon travail et pour son encouragement, ainsi que son esprit compréhensif tout au long de la thèse. Je le remercie pour tout son aide. Son enthousiasme et sa patience ont beaucoup facilité et agrémente mon travail.

Monsieur Pierre-Antoine CHAMPIN, Maitre de conférences à l'université Claude Bernard Lyon1, qui a consacré beaucoup de son temps pour des discussions, des conseils et des commentaires pertinentes sur le plan pratique et théorique. Il m'a énormément aidé, et ses remarques étaient très précieuses.

Je tiens également à remercier les membres du jury qui m'ont fait l'honneur de bien vouloir évaluer mon travail, et plus précisément :

Monsieur Tahar KIMOUR, Professeur à l'Université Badji-Mokhtar Annaba, pour l'honneur qu'il m'a fait, en acceptant la présidence de ce jury.

Monsieur Yacine LAFIFI, Professeur à l'université 8 Mai 1945 de Guelma, pour avoir accepté de juger le présent document. Je le remercie également pour ses commentaires pertinents qui ont permis d'améliorer la qualité de mon document

Monsieur Mahmoud BOUFAIDA, Professeur à l'université Mentouri Constantine, pour avoir accepté de juger le présent document, d'avoir accordé de son temps précieux pour se pencher sur mon travail et apporté des suggestions constructives pour l'amélioration de ce document.

Je tiens aussi à remercier :

Tous les membres de ma famille pour leur encouragement, je remercie spécialement

ma mère pour ses prières et je ne peux pas oublier à remercier fortement ma femme pour ses encouragements quotidiens et ses prières.

Tous mes proches et amis pour leurs aides et encouragements.

Tous les membres de l'équipe TWEAK à l'université de Lyon1 pour l'aide qui m'ont apporté au cours de mon stage de finalisation de thèse.

ملخص

يشهد مجال البيئات الرقمية تطورا مستمرا في العقود الأخيرة و بيئات التعلم ليست استثناء من ذلك. لقد أعطت هذه البيئات ومنذ ولادتها الكثير للتعلم الكلاسيكي. وإيماننا بمزاياها وإمكانياتها، كنا نميل إلى دراستها ومحاولة المساهمة فيها. هذه الأنظمة كما هو معروف في مجال الكمبيوتر، تدمج عدة أجزاء وتستخدم العديد من التخصصات. وفي إطار هذه الأطروحة نحن مهتمون بالجزء المتعلق بالسيطرة والمساعدة التربوية في هذه النظم. نعتقد في السياق نفسه أن المعلم الناجح هو الذي يسعى إلى فهم طلابه من أجل منحهم كل ما يحتاجونه بدقة وفعالية. إن الدراسات التي تركز على التنميط ونمذجة المتعلمين تتبنى نفس الفكرة، حيث أنها تسعى إلى توصيف المتعلمين لبناء السلوك المناسب في عملية التعلم التفاعلية.

مجال آخر يمكن إضافته إلى الأول هو مجال الآثار الرقمية. إن آثار التفاعل الرقمية هي بمثابة حاويات مهمة للمعرفة حيث أنها تساعد على فهم أفضل لسلوك المستخدمين وأنشطتهم. إن فهم سلوك المتعلم أثناء تعلمه لمنحه المساعدة الأكثر ملاءمة هو في الواقع الفكرة الأساسية لعملنا هذا. و لتكريس هذه الفكرة قدمنا مساهمتين.

المساهمة الأولى في هذه الأطروحة هي تطوير نظام لاستيراد آثار التفاعل الناتجة عن جلسات التعلم إلى نظام إدارة قواعد الآثار، وذلك للاستفادة من إمكانياته في التحليل والاستجواب غرض فهمها واستيعابها. ولقد قادنا هذا إلى اقتراح نظام لجمع آثار لنظم مختلفة باستخدام تقنية XPath. كما تم إجراء دراسة حول طبيعة الآثار الرقمية الموجودة في هذه النظم من أجل تحسين خياراتنا واقتراح نظام استيراد أكثر ملاءمة للآثار.

المساهمة الثانية هي عبارة عن طريقة جديدة لتفسير الآثار الرقمية وذلك لتغذية عمليات المساعدة. انتهجنا لذلك، البحث عن أنماط مفيدة من العناصر الملحوظة بمستوى أعلى من الدلالات. يتم هذا البحث عن الأنماط عن طريق تطبيق خوارزمية مطابقة هيكلية مع مقياس تشابه مختار بعناية.

الكلمات الدالة: المراقبة البيداغوجية، المساعدة، نظام تعلم رقمي، نظام تسيير آثار رقمية، آثار رقمية، نموذج آثار، جمع آثار، اكس باث، مطابقة، مقياس تشابه، نموذج متكرر

Abstract

The field of digital environments is constantly evolving in recent decades. Learning environments are not an exception. These environments and since their birth have given much to the classical learning. Believing in their advantages and possibilities, we have been inclined to study them and try to contribute to them. These systems as known in the computer science field, integrate several parts and make use of several disciplines. In this thesis, we are interested in the pedagogical control and assistance in these systems. In this context, we believe that the successful teacher is one who seeks to understand their students in order to give them exactly what they need with much relevance and effectiveness. Studies focusing on profiling and modeling learners adopt the same idea. They seek to characterize learners to build the appropriate behavior in an interactive learning process. Another new interesting domain we can add to the first is digital traces. In fact, traces of interaction are important containers of knowledge. They help to better understand the behavior of users and their activities. In the same line of ideas, understanding the behavior of the learner during his learning process to give him the most appropriate assistance is the basic idea of our work. To concretize this idea two contributions are made. The first contribution in this thesis is to import the resulting interaction traces of the learning sessions into a trace base management system to take advantage of its analysis and interrogation capabilities in order to gain better understanding of these traces. This led us to propose a system for collecting various system traces using XPath technology. A study on the nature of the existing traces was also made in order to refine our choices and to propose a more adapted importing of the traces. The second contribution is a new approach to interpret the traces as well to feed the assistance processes. The approach is based on looking for useful observed patterns with higher level of semantic. This pattern search is done by applying a structural matching algorithm with a precisely chosen measure of similarity.

Keywords : Educational control, Assistance, TEL, TBS, interaction trace, trace model, collection, xpath, mapping, similarity rule, frequent pattern

Résumé

Le domaine des environnements numériques connaît constamment une évolution rapide ces dernières décennies. Les environnements d'apprentissage ne sont pas une exception. Ces environnements et depuis leur naissance ont beaucoup donné à l'apprentissage classique. En croyant de leurs avantages et possibilités nous avons penché à les étudier et essayer d'y contribuer. Ces systèmes comme connus dans le domaine informatique intègrent plusieurs parties et faisant appel à plusieurs disciplines. Nous intéressons dans le cadre de cette thèse à la partie du contrôle et d'assistance pédagogique dans ces systèmes.

Dans ce même contexte, nous pensons que l'enseignant réussi est celui qui cherche à comprendre ses élèves afin de leur donner exactement ce dont ils ont besoin avec pertinence et efficacité. Les études s'intéressant au profilage et à la modélisation des apprenants adoptent la même idée. Elles cherchent à caractériser les apprenants pour leur construire le comportement adéquat dans un processus interactif d'apprentissage.

Un autre domaine d'actualité très répondu qui s'ajoute au premier est celui des traces numériques. Les traces d'interaction constituent d'importants conteneurs de connaissances. Elles permettent de mieux comprendre le comportement des utilisateurs et leurs activités. En effet, comprendre le comportement de l'apprenant durant son apprentissage pour lui donner l'assistance la plus adéquate est l'idée de base de notre travail. Pour concrétiser cette idée deux contributions sont faites.

La première contribution dans ce mémoire est d'importer les traces d'interaction résultantes des sessions d'apprentissage dans un système de gestion de base de traces pour profiter de ses possibilités d'analyse et d'interrogation en vue de leur compréhension. Ceci nous a conduit à proposer un système de collecte de trace de systèmes variés en utilisant la technologie XPath. Une étude sur la nature des traces existantes a été aussi faite afin d'affiner nos choix et de proposer une importation plus adaptée des traces.

La deuxième contribution est une nouvelle approche visant à interpréter les traces aussi en vue d'alimenter les processus d'assistance. L'approche consiste à chercher des patterns d'observés utiles ayant une sémantique de niveau plus élevé. Cette recherche de patterns est faite en appliquant une algorithmes d'appariement structurel avec une mesure de similarité choisie précieusement.

Mots Clés : Contrôle pédagogique, Assistance, EIAH, SBT, trace d'interaction, modèle de trace, collecte, xpath, mapping, appariement, mesure de similarité, motif fréquent

Table des matières

Avant-propos	i
Abstract	iv
Résumé	vi
Table des matières	ix
1 Introduction	1
1.1 Contexte	2
1.2 Question de recherche	4
1.3 Organisation de la thèse	7
2 État de l’art	9
2.1 Introduction	10
2.2 Assistance à l’utilisateur en milieu d’apprentissage	11
2.3 Traces d’interaction	19
2.4 Assistance à base de traces	24
2.5 Conclusion	26
3 Les traces dans les systèmes existants	27
3.1 Introduction	28
3.2 Quelques repères pour les EIAHs	28
3.3 Etude et analyse des traces existantes	30
3.4 Conclusion	47
4 Collecte et Importation des traces	49
4.1 Introduction	50
4.2 Processus Collecte et importation de traces	51
4.3 L’outil xCollector	57
4.4 Expérimentation	58
4.5 Conclusion	60

5	Interprétation des traces d'interaction	63
5.1	Introduction	64
5.2	Le simulateur CUSIM	65
5.3	Approche proposée	69
5.4	LogAnalyser, un système pour la recherche de patterns dans les fichiers Log	76
5.5	Conclusion	76
6	Conclusion	79
6.1	Synthèse	80
6.2	Perspectives et travaux futures	82
	Bibliography	93

Table des figures

2.1	Flux général de données dans un système assistant	15
2.2	Architecture d'un Système à Base de Trace [95]	23
2.3	Processus de collecte [95]	25
3.1	L'outil Performance Profiler de l'application Web DataShop	32
3.2	Interface du simulateur TELEOS	33
3.3	Interface du système APLUSIX	34
3.4	Page d'accueil de l'application web EDDBA	35
3.5	Interface d'Andes	36
3.6	Interface de SQL-Tutor	37
3.7	Interface du micromonde TPElec	38
3.8	Interface de COPEX-Chimie	39
3.9	L'étape d'adaptation dans AMBRE-add	40
3.10	Une fenetre de ColAT	41
3.11	Réponse à un exercice dans PEPITE	42
3.12	Un jeu dans le logiciel TRI	43
3.13	Environnement d'apprentissage collaboratif E-Lycée	44
3.14	Écran de choix d'activités de Tables au trésor	45
3.15	La plateforme Moodle	46
4.1	Processus de génération des collecteurs	52
4.2	Utilisation du collecteur	56
4.3	xCollector	59
5.1	Une interface de CUSIM et un exemple de log généré	66
5.2	Modèle de trace pour simulateur	67
5.3	Modèle de trace pour les observés généré par l'outil CUSIM	68
5.4	Approche proposée	69
5.5	Base de patterns	70
5.6	Exemple d'un appariement pour les traces de CUSIM	73
5.7	Visualisation des patterns et recherche des séquences fréquentes dans LogAnalyser	76

Liste des tableaux

4.1	Résultats du test des fonctionnalités essentielles de xCollector	61
5.1	Chevauchement des patterns	74

1

Introduction

De nos jours, les technologies numériques pénètrent tous les secteurs de notre vie et participent à la transformation des modes de production, de consommation, de communication, de circulation des savoirs et d'acquisition des connaissances. Le domaine de média par exemple a beaucoup élargie son spectre grâce aux dernières technologies utilisant le web comme support et les produits électroniques portables qui facilitent considérablement la production et l'accès à l'information. Le commerce est aujourd'hui au centre de ce changement car il est l'un des grands domaines bénéficiaires des technologies récentes. Le commerce n'a presque plus besoin de locaux physiques car la solution du web est devenue plus efficace et moins coûteuse. Beaucoup de technologies informatiques sont actuellement au service du marketing et du e-commerce. Les plateformes informatiques dédiées ainsi que le web, les réseaux sociaux et les outils d'analyse et de communication ont tous participé à l'évolution de tous les secteurs tels que : la culture, la science, l'économie, la politique, etc.

A son tour, le domaine d'enseignement et d'apprentissage a réservé sa place dans ce courant de rénovation. Aujourd'hui, les médias et même les individus parlent du e-learning et des plateformes d'apprentissage comme celle de Moodle ou Coursera ainsi que des sites destinés à l'apprentissage de langues comme celui de Duolingo. Dans un contexte plus restreint, la popularité croissante des plateformes entre autres du e-formation, de classes virtuelles, de télé-TP, de tuteur intelligent, etc., justifie le potentiel de ces technologies. selon l'avis de beaucoup de spécialistes, les Environnements Informatiques pour l'Apprentissage Humain (EIAH) basés sur la majorité des technologies

citées auparavant participent au développement du domaine d'apprentissage parallèlement avec les moyens résidentiels classiques. Cette participation permet de faciliter l'accès à l'apprentissage, propose une variété de contenus et omet les contraintes spatio-temporelles.

Les recherches dans le domaine de l'apprentissage par le numérique ne cessent d'avancer et d'aborder d'amples questions pour répondre aux intérêts et aux besoins croissants des utilisateurs. La modélisation du contenu et de l'apprenant, l'adaptation et la personnalisation de l'apprentissage ainsi que l'évaluation et la collaboration, sont quelques exemples de ces axes de recherche. Nous voulons dans notre thèse et à travers notre problématique aborder en particulier, l'un de ces axes qui est l'assistance (contrôle) dans les EIAH. Au cours de plusieurs années, une panoplie d'approches a été proposée pour répondre aux besoins d'assistance en milieu numérique d'apprentissage à savoir les systèmes conseillers, l'assistance contextuelle, etc. (voir chapitre 2 pour plus de détail). En effet, le travail que nous présentons est une contribution au développement de l'assistance en EIAH qui a été toujours considéré comme étant le maillon faible dans ces environnements, et qui s'ajoute à de nombreuses recherches ultérieures sur l'assistance en EIAH. Nous allons à notre tour l'aborder mais avec une vision et un contexte différents.

1.1 Contexte

Dans notre vie quotidienne, nous avons toujours besoin de l'aide, de conseil ou de l'assistance. Les premiers pas des enfants nécessitent naturellement les mains des parents pour les réussir. Au milieu du travail, pour déplacer un meuble de bureau dans une salle on aura peut être besoin d'appeler un collègue pour qu'il nous donne un coup de main. Un bug informatique simple, nous oblige parfois d'aller chercher ou demander de l'aide dans un forum pour le résoudre. Ainsi, l'aide ou l'assistance est un besoin naturel et continu dans notre vie. Par ailleurs, si les besoins de marcher chez l'enfant ou de déplacement de meuble sont claires, ce n'est pas toujours le cas pour tous les besoins à l'instar des bugs informatiques. La satisfaction de nos besoins en demandant de l'aide ou de l'assistance demande qu'ils soient clairs et explicites. Nous pouvons apprendre de ces exemples que plus nos besoins sont claires plus l'aide soit pertinente et efficace et inversement. Bien évidemment, la clarté et la compréhension du besoin est une information nécessaire mais pas suffisante pour la pertinence de l'aide.

Dans le cadre de l'enseignement, nous considérons que le parfait pédagogue est celui qui cherche constamment à comprendre ses élèves à travers ses facultés d'observation, d'interaction et d'expérience. Ceci va lui permettre de leur donner ce qu'ils ont besoin avec finesse et efficacité. Les études s'intéressant au profilage et au modélisation de l'apprenant adoptent la même idée. Ils cherchent à caractériser les apprenants afin de leur offrir le contenu, les activités et l'assistance adéquats dans un processus interactif d'apprentissage. L'enseignant qui sait bien interpréter le comportement de

ses apprenants et rassemble le maximum d'informations sur leurs états d'esprit est la seule personne (en excluant le cas d'hasard) pouvant leur choisir l'assistance la plus adéquate.

Dans cette thèse, nous nous intéressons au problème général de l'assistance à l'utilisateur dans un contexte particulier d'apprentissage : les EIAHs. Le rôle des outils d'assistance est de faciliter l'apprentissage et le partage des connaissances dans un environnement dynamique. Être en mesure d'offrir une assistance à diverses applications informatiques nécessite une représentation explicite des connaissances nécessaires pour exécuter un processus logiciel. De plus, pour soutenir le développement rapide du processus logiciel et les besoins en constante évolution des utilisateurs, un système d'assistance devrait disposer de mécanismes permettant de capturer des connaissances dynamiques de telle sorte qu'elles puissent être utilisées pour d'autres actions. Par conséquent, deux problèmes sont soulevés : A. Identifier les mécanismes de fourniture d'assistance et B. Découvrir les connaissances pour alimenter ces mécanismes. Bien évidemment, ces deux problèmes fonctionnent complémentaires et chacun est influencé par l'autre. Les mécanismes peuvent à la fois fournir de l'aide à l'utilisateur et enrichir la connaissance soit par le retour en arrière (feedback), soit simplement en faisant l'expérience de nouvelles situations. Nous nous intéressons plus précisément dans cette thèse, à la deuxième question et nous essayerons d'offrir des moyens aidant à la découverte de connaissances, à l'interprétation et à la compréhension pour alimenter les mécanismes offrant de l'assistance.

1.1.1 Définitions

Si l'on parle plutôt de l'assistance que de contrôle pédagogique, le terme qu'on est sensé utiliser dans ce manuscrit et qui figure d'ailleurs comme mot clef dans notre intitulé de thèse, c'est parce que c'est le synonyme le plus utilisé dans la littérature. Durant le parcours bibliographique sur l'assistance/contrôle pédagogique, beaucoup de similitudes entre les deux concepts ont été remarquées avec une divergence négligeable. Bien que les dictionnaires ont une autre opinion comme nous allons le voir ci-après, nous avons préféré d'utiliser tout le long de cette thèse le concept d'assistance pour exprimer simultanément les deux concepts. La définition que nous utilisons pour ces deux termes est : « L'assistance ou le contrôle pédagogique dans un milieu numérique d'apprentissage est l'aide proposée aux acteurs mobilisée par des moyens de capture et d'interprétation du comportement de l'acteur et de son contexte ». Cette définition est une sorte de jonction entre les deux questions liées à l'assistance précédemment invoquées. Elle peut être alors reformulée de la manière suivante : « Un processus cherchant à comprendre la situation d'apprentissage dans le but de proposer l'alternative d'aide la plus appropriée ».

Nous présentons maintenant un aperçu sur ce que disent les dictionnaires par rapport à ces deux concepts très proches.

LAROUSSE : Assistance : Action d'assister quelqu'un, de l'aider dans une difficulté, un danger : Prêter assistance à un blessé. Contrôle : Action de contrôler quelque chose, quelqu'un, de vérifier leur état ou leur situation au regard d'une norme : Contrôle des touristes à la douane. Tour de contrôle d'un aérodrome. Contrôle médical. Action, fait de contrôler quelque chose, un groupe, d'avoir le pouvoir de les diriger : Perdre le contrôle de son véhicule. Avoir le contrôle d'un territoire. Aide : Action d'aider quelqu'un, de lui donner une assistance momentanée ; appui, soutien : Avec votre aide, je réussirai.

Le DICTIONNAIRE : Assistance : Aide, secours. Contrôle : Action de mener, diriger, gouverner. Aide : Action d'apporter son soutien physiquement, verbalement, financièrement ou de tout autre manière.

LEXILOGOS : Assistance : Action d'aider, de secourir quelqu'un Contrôle : Vérification portant sur des choses en vue d'examiner si elles remplissent les conditions demandées. État de domination morale, matérielle ou politique dans lequel se trouve soumis un pays, une région, une personne. Aide : Action d'aider quelqu'un, concours que l'on prête, soutien moral ou secours matériel que l'on apporte.

L'INTERNAUTE : Assistance : Fait de venir en aide de secourir quelqu'un en difficulté, de venir dépanner un objet tombé en panne. Contrôle : Vérification, surveillance attentive. Maîtrise, garder la commande de quelque chose. Aide : Assistance, appui, action d'aider. Secours financier accordé aux plus démunis ou victimes d'un événement.

REVERSO : Assistance : aide, secours, action pour venir en aide. secours, appui, subvention, soutien, réconfort, charité, renfort, concours, aide, protection, rescousse. Contrôle : secours, appui, subvention, soutien, réconfort, charité, renfort, concours, aide, protection, rescousse. Vérification, pointage, supervision, inspection, expertise. Aide : Assistance, soutien. Secours, renfort, rescousse, assistance, guide en ligne, appui, solitude, abandon.

1.2 Question de recherche

A l'ère de l'information, l'intérêt est de plus en plus centré sur l'information qui représente le chemin qui mène vers l'extraction de la connaissance. Par exemple, Les logs (fichiers historiques de l'exécution des serveurs web) étaient considérés comme support d'informations techniques liées au trafic d'informations dans les serveurs web destinés aux administrateurs de réseaux. L'évolution des serveurs, du contenu des logs et des outils d'analyse ont motivé aujourd'hui beaucoup d'acteurs, chercheurs et développeurs, à les exploiter. En plus de ce qui est offert par les logs, les traces d'activités sont encore plus riches et complexes en contenu et leurs outils sont plus théorisés et intéressants. A l'instar de plusieurs recherches, notre travail est centré sur la notion de traces d'activités. Ces traces produites par toute utilisation d'un système informatique et qui sont considérées comme source de connaissances, peuvent en conséquence être

exploitées en retour pour analyser, soutenir et assister l'activité tracée. Dans le cadre des EIAHs, nous nous intéressons aux traces d'activité de l'apprenant qui utilise un EIAH afin d'élaborer un profil de l'apprenant, de lui proposer de l'assistance appropriée ou de l'évaluer, etc. Les EIAHs existants permettent en général d'obtenir des traces de l'activité des apprenants.

Que ce soient d'interaction ou d'activité, les traces sont déterminées comme étant des inscriptions d'informations liées à l'interaction/l'activité des utilisateurs avec un système informatique. A cause de leur aspect porteur de connaissances, les traces se considèrent comme un nouveau paradigme autour duquel plusieurs axes sont traités à savoir : la collecte de traces, taxonomie et modélisation de traces, la transformation de traces, l'interprétation de traces, stockage, requêtage, visualisation, etc.

La réutilisation de traces pour une activité donnée nécessite généralement une phase d'analyse et de transformation de la trace collectée afin de proposer une interprétation fidèle à l'utilisateur. Néanmoins, l'analyse des traces n'est pas un processus simple. Imaginez une simple trace "Une trace de copier-coller" par exemple, à première vue elle semble très facile à manipuler. Mais en réalité, beaucoup d'informations sont à identifier. Nous devons déterminer l'acteur (utilisateur), le temps, la source et la destination du processus "copier". Selon que le type d'objet à copier est graphique ou textuel d'autres informations peuvent être considérées. Par conséquent, pour une activité complexe, la tâche d'analyse ou de transformation de traces devient plus compliqué et nécessite une profonde manipulation.

Les traces d'activité ont été traitées comme sources riches en connaissances pouvant alimenter d'autres modules. Comme le confirment plusieurs recherches et applications, les traces ont beaucoup contribué à l'amélioration de la qualité de l'assistance proposée dans les EIAHs. En effet, plusieurs approches d'assistance font appel aux traces pour modéliser le comportement et le contexte de l'apprentissage. Ces deux types de modélisation représentent deux briques importantes dans un module d'assistance efficace. Nous allons justement dans cette thèse se concentrer sur cet aspect particulier d'utilisation des traces : l'analyse et l'interprétation des traces afin d'aider le module d'assistance à calculer son choix en termes d'interventions (réactives ou proactives).

Après cette introduction rapide sur les traces d'activité et leur potentiel notamment pour l'assistance en EIAH, nous annonçons maintenant notre problématique de recherche liée à ces deux concepts. La question générale qui nous intéresse est "Comment exploiter ces traces d'activité riches en informations afin d'assurer une assistance efficace dans les EIAHs" ? Nous devrions rappeler ici que le processus d'assistance est une jonction de deux phases : (A) L'interprétation et la compréhension du comportement et du contexte d'apprentissage, et (B) Le calcul et le choix de la modalité d'intervention de l'assistance. La phase qui va être abordée dans cette thèse est la phase (A) et notre effort va être centré sur l'études des traces, leur analyse, transformation et interprétation ainsi que d'autres problèmes connexes. Nous pouvons alors reformuler notre précédente question comme suit : "Comment arriver à comprendre le contexte ou le

comportement des apprenants interagissant avec un EIAH à partir de leurs traces d'interaction?" sachant que ces traces sont généralement de nature technique et elles sont très variées en contenu et en format. Il est évident alors que ces caractéristiques vont compliquer tout traitement adressé aux traces.

Au sein du laboratoire LIRIS (France) a été définie la notion de Système à Base de Traces (SBT), qui permet de stocker, transformer et interroger les traces de l'activité. Les traces manipulées au sein du SBT sont des traces modélisées, dans le sens où elles sont conformes à un modèle explicite du système informatique tracé. Ses systèmes sont donc un moyen très utile qui pourrait nous amener à la compréhension des traces. Selon que nous faisons appel aux SBT ou pas, deux pistes d'interprétation de traces sont envisageables menant à deux questions secondaires liées à cette thèse. La première traite le problème de la collecte de traces en vue d'une importation dans un SBT. Dans ce cas, le passage par un SBT va faire abstraction de l'analyse directe des traces. Le deuxième choix cible directement des traces brutes et essaye à travers de nouvelles techniques de donner des interprétations aux activités des apprenants en fouillant dans leurs traces.

La première question de recherche qui se pose est : comment permettre à un concepteur d'EIAH, qui possède des traces d'activités issues de cet EIAH, de les importer en tant que traces modélisées dans le kTBS (kernel for Trace Based System), afin de pouvoir profiter de ses fonctionnalités de requêtage et de transformation ? kTBS est un système qui a été développé au sein du laboratoire LIRIS pour mettre en œuvre la notion de système à base de traces. Il s'agit alors, pour chaque EIAH, de générer ce que nous appelons un collecteur. Ce collecteur prend en entrée les traces d'activités existantes de l'EIAH, qui peuvent être des fichiers XML, des fichiers texte structurés ou des bases de données. Le modèle de ces traces étant souvent implicite. Ce collecteur doit fournir en sortie des traces conformes au modèle défini pour cet EIAH au sein du kTBS, modèle exprimé dans le langage du kTBS fondé sur RDF.

La problématique de ce travail de recherche relève donc de l'acquisition des connaissances, puisqu'il s'agit de proposer au concepteur d'un EIAH un moyen d'explicitier la correspondance entre les traces dont il dispose et le modèle de trace qu'il a défini au sein du kTBS. Ces liens de correspondance devront permettre à un système à concevoir, de générer automatiquement le collecteur qui effectuera le travail de réécriture des traces, afin de les importer dans le kTBS.

Une question sous-jacente à cette problématique est celle de la genericité des interfaces que l'on proposera à l'utilisateur d'une part, et des processus permettant la génération des collecteurs d'autre part, sachant que les traces d'EIAH existantes sont relativement variées, à la fois dans leur contenu et dans leur représentation technique.

Etant donné que les traces sont considérées comme source d'information importante pour les systèmes informatiques qui les produisent et les intègrent, plusieurs recherches ont été faites sur leur interprétation pour la prise de décision. Décider de la manière d'assistance n'est pas une exception, en effet, les traces peuvent dans ce cas

conduire à bien concevoir l'activité des apprenants. Durant une session d'apprentissage nous pouvons collecter beaucoup d'informations expliquant ce qui se passe du genre : activité en cours, informations manipulées, informations produites par l'apprenant, les communications inter-apprenants, ainsi que beaucoup d'autres. Toutes ces informations peuvent alimenter l'assistance qui est dans notre contexte le preneur de décision.

Les traces d'activité sont le plus souvent de nature technique loin du langage des analystes et des enseignants ce qui justifie la nécessité de les interpréter et de les réécrire. La question de recherche qui se pose est "Comment réécrire et transformer les traces d'activité et passer de leur langage technique de bas niveau à un langage facile et compréhensible d'un niveau d'abstraction plus élevé destiné aux analystes/enseignants?" Il s'agit alors de transformer les traces brutes produites par les EIAH en traces plus simples et plus utiles pour les analystes/enseignants.

Des questions sous-jacentes au processus de transformation de traces sont à traiter. La généralité des solutions que nous proposons est l'une de ces questions, une étude approfondie sur les contenus des traces pourrait en donner une réponse. Le stockage, la modélisation et le prétraitement des traces brutes et transformées, sont d'autres questions qui se soulèvent et que nous allons essayer de résoudre dans nos investigations.

Nous voulons maintenant récapituler cette discussion autour de notre problématique en rappelant que notre thématique générale est le contrôle pédagogique dans les EIAHs. L'avancée et les possibilités intéressantes qu'offrent les traces d'interactions en termes d'interprétation et d'explication des apprentissages nous ont poussé à les exploiter pour répondre à notre question "Comment exploiter ces traces d'activité riches en informations afin d'assurer une assistance efficace dans le contexte des EIAHs?" La réponse à cette question dépend des moyens à disposition. Autrement dit, la réponse est définie suivant l'utilisation ou pas des SBTs ces systèmes dédiés au traitement de traces. Notre problème a été ainsi détaillé en deux questions secondaires, à savoir, la collecte et l'interprétation de traces. Ces deux questions sont rassemblées avec d'autres éléments intrinsèques dans la question suivante : Quelles traces utiliser, comment les acquérir et collecter et quel moyen utiliser pour les interpréter?

1.3 Organisation de la thèse

Dans ce chapitre, nous avons présenté une introduction générale de notre travail par le passage des nouvelles technologies de l'information et de la télécommunication, ses potentiels et possibilités ainsi que ses influences sur les enjeux actuels. Notre contexte a été aussi bien clarifié dans ce chapitre en parlant sur les environnements informatiques pour l'apprentissage humain, l'assistance et son importance dans ses systèmes ainsi que les avantages d'utiliser les technologies de traces d'interaction dans le même contexte. Dans ce chapitre, nous avons présenté notre problématique de recherche, ses

implications ainsi que certains choix. Beaucoup de détails sur cette problématique seront donnés dans les chapitres qui suivent.

Dans le chapitre suivant (chapitre 02) nous présenterons un état de l'art sur l'assistance des utilisateurs dans les systèmes informatiques, ses besoins, ses moyens et ses approches. Nous présenterons également quelques expériences de systèmes d'assistance existants et nous parlerons plus particulièrement en détail sur l'assistance basée sur les traces d'interaction. Le concept de trace sera aussi abordé dans ce chapitre en parlant sur son existant dans la littérature. Nous expliquerons par la suite quelques concepts clés en relation avec la modélisation, la collecte de traces et les systèmes à base de traces.

Le chapitre 03 présentera notre première investigation de recherche sous forme d'une étude faite sur des environnements d'apprentissage et des logiciels informatiques en vue de décrire les traces d'utilisation qu'ils produisent. Une telle étude va nous permettre de connaître et de comprendre de plus près la nature et les formats utilisés pour les traces des systèmes existants. Les résultats de cette étude vont nous permettre de tailler nos choix et de proposer des solutions génériques autant que possible.

Le chapitre 04 présentera notre deuxième investigation qui traite le problème de collecte de traces dans un contexte précis qui est celui des systèmes à base de trace et plus précisément le système KTBS développé par l'équipe SILEX du laboratoire LIRIS. Nous expliquerons dans ce chapitre chacune des étapes de notre processus de collecte pour les trois formats de traces considérés. Nous présenterons à la fin de chapitre notre outil de collecte xCollector avec une expérimentation auprès de quelques utilisateurs.

Dans le cinquième chapitre nous présenterons notre troisième investigation qui est une approche basée sur l'appariement structurel pour interpréter des traces d'interaction. Nous expliquerons également la mesure de similarité que nous avons adoptée dans notre algorithme d'appariement. Un algorithme complémentaire de recherche de séquences fréquentes sera expliqué. Nous terminerons ce chapitre par la présentation d'un logiciel concrétisant notre approche et ce pour montrer son efficacité et faisabilité.

Le dernier chapitre (chapitre conclusion) clôturera notre manuscrit de thèse par la confrontation de la problématique soulevée initialement aux différentes investigations réalisées. Nous discuterons aussi dans ce chapitre les problèmes rencontrés, les pistes à franchir et à éviter dans certaines situations et nous terminerons le chapitre par un ensemble de perspectives pouvant enrichir cette recherche ou plutôt guider d'autres chercheurs dans le domaine.

2

État de l'art

Plan du chapitre

Table des matières	ix
------------------------------	----

2.1 Introduction

Avec le développement de nouvelles technologies de l'information, les applications et les outils sont devenus très complexes. Par conséquent, il est essentiel de fournir aux utilisateurs une assistance adaptée pour les aider à utiliser ces applications et outils. Les approches d'assistance classiques fournissent souvent une assistance statique pré-calculée. Toutefois, le contexte doit être pris en compte afin de fournir aux utilisateurs des informations utiles au bon moment et dans la bonne situation. Plusieurs approches ont été utilisées pour modéliser le contexte en l'occurrence les ontologies et les traces d'interaction.

Les traces d'interaction sont largement traitées aujourd'hui comme source prometteuse de connaissance. Beaucoup de recherches montrent que la capture des connaissances gravitant dans les traces peut supporter considérablement la création d'un assistant dynamique. De nos jours, de nombreuses applications collectent des fichiers journaux (logs) ainsi que des traces d'interaction d'utilisateurs de différents types, les logs par exemple sont les plus simples. Ils contiennent des données brutes collectées dans un but spécifique. Un exemple typique est celui du log d'un serveur Web qui conserve l'historique des demandes de pages. L'accès à ces fichiers est généralement restreint aux webmasters et aux administrateurs. Avec les outils d'analyse des fichiers log, plusieurs informations peuvent être extraites tel que la provenance des visiteurs, leur fréquence de consultation et la manière dont ils naviguent dans un site. Les traces d'interaction sont une forme de trace plus complexe. Une trace d'interaction est un enregistrement riche des actions effectuées par un utilisateur sur un système. Ces traces d'interaction permettent de capturer les expériences des utilisateurs dans des environnements numériques. L'analyse statistique des traces peut plutôt conduire à comprendre les comportements des utilisateurs et leurs activités. L'analyse des traces peut également être utilisé pour détecter des modèles utiles ou des modèles d'erreur dans le comportement des usagers.

Le but de cette introduction n'est pas seulement de préparer le lecteur avant voir de proche et en détail les deux concepts d'assistance et de traces, mais aussi pour mettre en valeur la relation entre eux. Les systèmes d'assistance comptent beaucoup sur l'exploitation et l'analyse des traces d'interaction et les considèrent comme support et source de connaissance très riche. Nous présentons respectivement le concept d'assistance et de traces d'interaction dans les deux sections suivantes. La section 4 expliquera les systèmes d'assistance basés sur les traces d'interaction et montre ainsi le couplage entre ces deux concepts, la section 5 sera la conclusion de notre chapitre.

2.2 Assistance à l'utilisateur en milieu d'apprentissage

Les recherches en EIAH ont conduit à une somme d'outils intéressants. Malgré cela, l'usage de ces EIAH reste limité. La complexité et les interfaces souvent chargées des outils destinés aux enseignants font que ces derniers peuvent en effet se sentir dépassés et renoncer à les utiliser. L'adjonction d'un système d'assistance à ces logiciels complexes permet de les prendre en main et de les utiliser plus facilement, tout en évitant une sous-exploitation qui limiterait leur intérêt et leur richesse. Les enseignants désireux d'intégrer un EIAH dans leurs pratiques préfèrent parfois choisir un logiciel éducatif simple qui correspond partiellement à leurs besoins et qui leur propose peu ou pas de possibilités d'adaptation. Dans ce cas, la mise en place d'un système d'assistance personnalisée destiné aux apprenants est une solution pertinente pour permettre aux enseignants d'adapter ces outils à leurs pratiques et à leurs intentions pédagogiques [46].

L'assistance à l'utilisateur, ou aide, englobe tous les moyens mis en œuvre pour éviter qu'un utilisateur sous-exploite une application, ou doive se tourner vers un autre utilisateur, plus expert, pour l'aider à surmonter ses difficultés. Du point de vue des sciences cognitives, Gapenne [42] définit l'aide comme étant *une relation, asymétrique et instrumentée, entre une personne ayant un projet d'action (souhaité ou suggéré voire imposé) dont les modalités de réalisation sont ignorées (ou oubliées) et une technologie censée rendre explicite ces modalités, d'une façon telle qu'elles soient appropriables par la personne sollicitant l'aide*. Gapenne a cité quatre modalités pour le couplage humain/technologie :

- *la substitution* : lorsque la technologie prend en charge de manière autonome la totalité ou une partie d'une tâche ;
- *la suppléance* : lorsque l'utilisation de la technologie augmente les possibilités d'action de l'utilisateur et que l'on peut observer de nouveaux schémas ou invariants ;
- *l'assistance* : lorsque la technologie n'est pas cruciale pour l'activité principale. Le rôle principal de la technologie est de faciliter ou d'améliorer l'utilisation de l'outil principal ;
- *le support* : lorsque la technologie permet de supporter l'appropriation et l'utilisation d'un nouveau schéma par l'homme.

Cette section vise à clarifier ces concepts d'aide et d'assistance. Nous allons ainsi essayer de les expliquer de plusieurs points de vues tout en invoquant les questions inhérentes les plus critiques.

2.2.1 Approches d'assistance

Selon [46], les moyens d'assistance permettant de répondre aux besoins des utilisateurs les plus variés sont nombreux et peuvent être mis en œuvre grâce à différentes

approches d'assistance qui peuvent être complémentaires. Nous présentons dans cette section une vue d'ensemble de ces approches.

Un manuel d'aide rassemble un ensemble d'instructions ou d'informations concernant un produit, souvent présenté sous forme textuelle mais aussi en multimédia. Ces manuels sont fréquemment utilisés pour l'assistance aux utilisateurs de logiciels. Souvent accessibles en ligne pour les applications informatiques, ils peuvent être consultés à tout moment par l'utilisateur selon ses besoins. Les manuels d'aide, souvent adaptés aux utilisateurs standards, ne conviennent pas toujours pour des utilisateurs novices ou experts. En effet, un utilisateur novice peut ne pas se rendre compte qu'il a besoin d'aide, ou bien être incapable d'identifier ou de formuler son besoin d'aide, ce qui limite l'efficacité du manuel d'aide. Par ailleurs, un utilisateur expérimenté peut trouver fastidieuse la recherche d'une information qui lui manque parmi de nombreuses informations qu'il connaît déjà.

L'aide contextualisée constitue une réponse à ces problèmes. Il s'agit en effet d'une assistance directement liée au contexte dans lequel se trouve un utilisateur [25]. Ainsi, une aide contextualisée sous forme de messages d'assistance pour les apprenants peuvent par exemple être intégrée dans des scénarios pédagogiques si l'enseignant le souhaite.

Un système conseiller est un système qui propose une aide active à l'utilisateur d'un logiciel particulier, aide fondée sur une analyse des actions et des productions de l'utilisateur [83]. Les systèmes conseillers proposent souvent à l'utilisateur de choisir entre plusieurs modes d'assistance qui peuvent notamment être catégorisés par le degré d'interventionnisme du système d'assistance.

Quant aux *agents conversationnels*, ce sont des personnifications de la fonction d'assistance qui proposent une aide à l'utilisateur. Leur but est d'inciter au dialogue et de le faciliter par effet de sympathie [64]. Ils peuvent avoir plusieurs apparences textuelle ou graphique. Les agents graphiques peuvent être animés et exprimer des émotions (empathie, surprise, mécontentement, etc.), ce qui facilite la communication avec l'utilisateur et augmente la crédibilité de l'agent. Plusieurs agents conversationnels ont été proposés comme le célèbre Clippy de Microsoft et d'autres capables même de comprendre des questions en langage naturel oral.

Un système de recommandations est un système qui produit des recommandations individualisées ou qui a pour effet de guider l'utilisateur de manière personnalisée vers des objets utiles ou intéressants parmi un large choix d'options possibles [23]. Si les systèmes de recommandations sont souvent mis en œuvre dans des applications commerciales sur internet, ils peuvent également aider un utilisateur à gagner du temps en le guidant vers des choix pertinents.

Un tutoriel est un outil pédagogique permettant à un utilisateur de se former de manière autonome. Un tutoriel peut se présenter sous diverses formes (application, vidéo, texte...) et contient des explications détaillées pas à pas. Ils sont fréquemment utilisés pour assister les utilisateurs, en particulier pour des logiciels grand public. Un tutoriel

peut être intégré dans une application ou être indépendant : il existe ainsi de nombreux tutoriels en ligne, qu'ils soient créés par les concepteurs de l'application concernée ou par des utilisateurs désireux de faire partager leur expérience.

Une interface adaptative est capable d'adapter son comportement aux besoins, capacités et préférences de l'utilisateur courant pendant l'interaction, grâce à ses capacités de perception et d'interprétation de l'interaction et de son contexte [97]. Elles permettent d'aider l'utilisateur en personnalisant l'interface de l'application, comme le font par exemple, les menus adaptatifs d'Office 2003.

La réutilisation de l'expérience peut constituer une approche d'assistance en permettant à un utilisateur de réutiliser des actions passées [111]. Ces actions peuvent avoir été effectuées lors d'une utilisation antérieure de l'application, soit par l'utilisateur lui-même, soit par un autre utilisateur dans une situation analogue.

Les communautés de pratiques désignent un ensemble de personnes qui partagent des pratiques communes, rassemblées par des relations informelles par une expertise partagée ou par un centre d'intérêt commun [106]. Les communautés de pratiques peuvent constituer une approche d'assistance en regroupant les utilisateurs d'une même application et en leur permettant de s'entre-aider dans plusieurs domaines.

2.2.2 Moyens d'assistance

Les besoins d'assistance sont divers et dépendent d'une part de l'application en elle-même, de sa complexité, de sa richesse, et d'autre part de l'utilisateur, de son niveau de maîtrise de l'application et des objectifs pour lesquels il utilise cette application. Afin de répondre à la variété de ces besoins, qu'ils soient explicitement identifiés ou non, il existe différents moyens que nous discuterons dans ce qui suit. [46] regroupent les moyens d'assistance observés dans les systèmes existants en quatre catégories : les messages, les exemples, les modifications de l'interface et la création automatisée.

Les messages d'assistance permettent de communiquer des informations à l'utilisateur. Ils sont très utilisés dans de nombreuses applications et constituent un moyen à la fois simple et efficace d'assister l'utilisateur. Ils sont souvent proposés sous forme de fenêtres pop-up, de bulles d'aides ou par un compagnon s'adressant directement à l'utilisateur. Les messages d'aide peuvent être distingués selon trois types de contenus : du texte (recommandations, explications), des liens (vers une page web), et des raccourcis vers une fonctionnalité.

Les aides de type *exemple* permettent de réduire la distance entre l'explication et la tâche concrète de l'utilisateur. Il peut s'agir d'exemples fournis par l'application dans le but d'illustrer les possibilités de l'application, ou d'un aperçu du travail déjà réalisé par l'utilisateur. Il peut s'agir de démonstrations, par exemple sous forme de vidéos, expliquant à l'utilisateur comment réaliser une tâche ou prendre en main l'application, comme cela est proposé dans l'aide en ligne d'Office 2010.

L'assistance par la *modification de l'interface* consiste à effectuer des changements directement sur l'écran en jeu. Les modifications peuvent être de type masquage, pour griser une fonctionnalité indisponible par exemple, ou au contraire de type mise en valeur, par exemple pour indiquer à l'utilisateur qu'il n'a pas rempli le champ d'un formulaire ou pour mettre en valeur une fonctionnalité qui pourrait être utile à l'utilisateur.

La création automatisée, parfois qualifiée de substitution, permet de simplifier la tâche de l'utilisateur et de lui faire gagner du temps en réalisant tout ou partie de la tâche de l'utilisateur à l'aide de valeurs par défaut, de patrons ou de modèles à personnaliser. Les valeurs par défaut permettent de pré-remplir automatiquement les champs d'un formulaire. Elles peuvent être prédéfinies par le système, ou proposées à l'utilisateur en fonction des valeurs qu'il utilise habituellement, ou encore en fonction des valeurs utilisées par d'autres utilisateurs ayant des besoins, des objectifs ou des caractéristiques proches de ceux de l'utilisateur. Un patron est une solution, qui peut être partielle, à un problème récurrent dans un contexte donné, établie par un concepteur ou un utilisateur averti. Il est exprimé par un ensemble de spécifications qui sont exploitées par l'application lors de la réalisation de la tâche. Un modèle est quant à lui un objet du même type que l'objet à produire par la tâche en cours. Il a été défini préalablement par un concepteur ou un utilisateur averti, à des fins de réutilisation.

2.2.3 Besoins d'assistance

[46] identifie différents besoins d'assistance des utilisateurs, qu'ils ont ensuite regroupés en catégories puis hiérarchisés. Il s'agit de trois principaux besoins d'assistance : la découverte du système, la réalisation d'une tâche et l'amélioration de la pratique.

Les besoins d'assistance pour *la découverte du système* regroupent les besoins de compréhension générale du système et de découverte d'une fonctionnalité particulière, qui concernent principalement les utilisateurs novices : à quoi sert le système, quels sont les termes et objets manipulés dans le système et comment peut-on les utiliser, quelles sont les fonctionnalités proposées par le système et le cas échéant quelles sont les étapes nécessaires à la réalisation d'une tâche.

Les besoins d'assistance pour *la réalisation d'une tâche* concernent tous les utilisateurs, qu'ils soient ou non novices. Parmi ces besoins, la découverte d'une nouvelle fonctionnalité peut être nécessaire pour un utilisateur lorsqu'il effectue, pour la première fois notamment, une tâche qui requiert l'utilisation d'une ou plusieurs fonctionnalités du système. De plus, lorsqu'une tâche est complexe ou longue à réaliser, un suivi de la tâche peut aider l'utilisateur à mieux comprendre ce qu'il fait. Une projection peut aider l'utilisateur à mieux comprendre ce qu'il a fait, et éventuellement à repérer ce qu'il doit rectifier dans son travail pour obtenir le résultat souhaité. Cette projection peut notamment se faire sous la forme d'un aperçu du travail réalisé par l'utilisateur, ou d'une

instanciation de son travail sur un exemple. Parmi les besoins qui concernent le suivi de la tâche, une validation peut être nécessaire pour certaines tâches, pour confirmer à l'utilisateur que ce qu'il a fait est correct, ou au contraire pour mettre en évidence les problèmes ou les incohérences de ses productions. Parmi les besoins qui concernent le suivi de la tâche, un bilan de ce qui a été fait et de ce qu'il reste à faire peut être utile à l'utilisateur. Par ailleurs, la réalisation d'une tâche peut entraîner un besoin de guidage, afin notamment d'aider l'utilisateur à comprendre ce qu'il doit ou peut faire à chaque instant. Enfin, Il est possible de faciliter la tâche de l'utilisateur, en pré-réalisant cette tâche partiellement ou totalement, tout en lui permettant de modifier ou compléter les propositions qui lui sont faites.

Les besoins d'assistance relatifs à *l'amélioration de la pratique* concernent tous les utilisateurs. L'amélioration peut porter d'une part sur l'enrichissement de la pratique de l'utilisateur, notamment en mettant en évidence des options ou fonctionnalités offertes par le système et non utilisées, et d'autre part sur temps de réalisation d'une tâche, par l'automatisation de certaines parties de la réalisation d'une tâche par exemple.

2.2.4 Dimensions d'assistance

Bien que la gamme de systèmes d'assistance possibles soit très large, les groupes de systèmes d'assistance peuvent être distingués par leurs caractéristiques d'offre d'informations et d'extraction de données. Selon [87] le flux de données général dans un système d'assistance est illustré à la figure suivante.

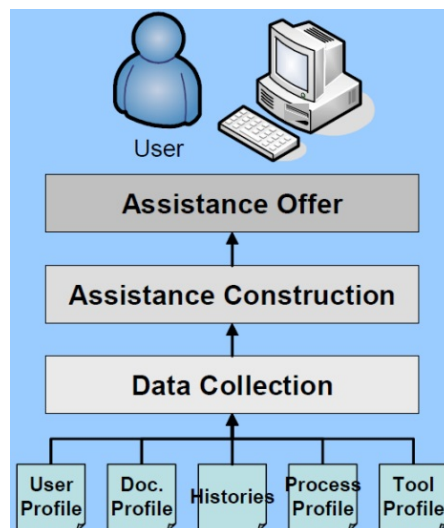


FIGURE 2.1 – Flux général de données dans un système assistant

Les informations du contenu du travail, du processus, du statut de l'outil et des informations sur l'utilisateur sont collectées et mises à disposition dans un format unifié.

Ces informations sont utilisées par les algorithmes (approches) d'assistance pour produire des informations spécifiques à un contexte proposées à l'utilisateur. L'assistance fournie à l'utilisateur est basée sur l'algorithme de construction. Bien que le résultat de cet algorithme soit fixe, la méthode de présentation peut être différenciée par les caractéristiques suivantes [87] :

- *Quand assister ?* si chaque clic de l'utilisateur indique une action potentielle, la question est de savoir quand l'utilisateur doit être assisté. L'assistance peut être générée de manière proactive, pendant ou après les actions, et présentée à ou sur demande.
- *Comment assister ?* étant donné que les ordinateurs modernes présentent souvent des environnements de travail multimédias, la forme de média utilisée par l'assistance peut être différenciée. Actuellement, l'assistance peut être présentée textuellement ou sous forme de multimédia.
- *Où assister ?* les informations fournies par le système d'assistance peuvent être regroupées dans des info-bulles, des fenêtres contextuelles, des tableaux, des effets sonores spécifiques, des effets clignotants, des barres latérales d'un document ou des espaces marqués spécifiques. De plus, il peut être présenté dans l'outil actif, un outil tiers spécifique ou dans le système d'exploitation lui-même.
- *Pourquoi assister ?* Il peut exister une lacune de compétences dans le profil de l'utilisateur, une étape de processus complexe peut être avancée, de nouvelles fonctionnalités d'outil ont été intégrées lors d'une mise à jour ou un algorithme généralement sujet aux erreurs est en cours de développement.

Même si un outil d'assistance a au moins une caractéristique fixe, il est également possible que le système décide lui-même quoi, quand et comment l'information doit être présentée à l'utilisateur. Cette décision peut être influencée par des informations sur les raisons pour lesquelles et pour qui une assistance a été générée. L'assistance est disponible dans toutes les tailles et tous les goûts - d'une simple explication d'outil à une offre étendue d'apprentissage en ligne. Bien que la fonctionnalité ou le résultat en soi soit difficile à classer, les algorithmes d'assistance peuvent être caractérisés comme suit :

- *Assistance pour qui ?* Qui devrait être aidé avec les informations construites ? Selon le profil de l'utilisateur, les résultats doivent être personnalisés ou adaptés au niveau d'expertise.
- *Assistance sur quoi ?* quel type d'objet devrait être enrichi d'informations d'assistance ? S'agit-il d'un document d'exigences, d'un outil de test, d'un processus ou d'un modèle d'activité, de connaissances de base générales, d'informations sur des experts, etc. ?
- *Assistance dans quel processus ?* Le processus ou l'activité dans laquelle l'utilisateur est actuellement impliqué pourrait induire un besoin particulier d'assistance. Par exemple, un programmeur qui est un développeur de logiciel a d'autres besoins d'assistance qu'un testeur ou un inspecteur qui examine le

- même logiciel.
- *Assistance dans quel outil ?* Si les informations sur le processus ne sont pas disponibles, le contexte d'environnement de l'outil peut également être utilisé afin d'optimiser l'assistance. Un outil de test en cours peut être un indicateur du processus actuel ; L'utilisation d'un éditeur de texte pour un document d'exigences peut impliquer des connaissances manquantes ; et un environnement de codage mis à jour peut faire allusion à de nouvelles fonctions inconnues de l'utilisateur.

2.2.5 Modalités d'intervention dans les systèmes d'assistance

Comme déjà mentionné, les besoins, contextes et moyens d'assistance sont variés ce qui implique aussi la nécessité d'adopter des degrés et modalités différentes et d'intervenir parfois des acteurs supplémentaires pour assurer une assistance pertinente. Nous voulons présenté ici quelques modalités citées dans [82].

- *Assistance réactive* : ce mode figure dans les systèmes qui se contentent de donner une simple rétroaction à l'apprenant. C'est le cas notamment de la plupart des progiciels, des hypermédias et de certains systèmes experts. L'apprenant utilise les outils du système pour réaliser une tâche ou résoudre un problème. Le système affiche certaines conséquences des actions de l'apprenant d'une façon qui devrait l'aider à cheminer vers une solution.
- *Assistance semi-proactive* : il s'agit ici d'un niveau interventionniste moyen consiste à prévoir une aide interactive intelligente. Un tel mécanisme affichera, sur demande de l'apprenant, une explication relative à la composante de l'interface où il se trouve ou à l'outil qu'il utilise, le plus possible en relation avec la tâche qu'il est en train de réaliser. Un bel exemple d'une telle approche se retrouve dans le module d'explication d'un système expert.
- *Assistance proactive* : le niveau interventionniste est encore plus élevé comme le cas dans les systèmes *coach* ou les conseillers actifs. Dans tels systèmes, l'apprenant n'obtient pas seulement sur demande une aide ciblée sur ses activités, le système peut également décider d'intervenir pour afficher un conseil lorsqu'il lui semble que l'apprenant en a besoin pour réussir la tâche en cours. Cependant, le conseil n'est pas impératif ici et c'est toujours à l'apprenant de le suivre ou non.
- *Assistance tutorielle* : dans les systèmes tutoriels intelligents l'initiative est entièrement laissée au système qui exerce un guidage de l'apprenant de type tutorat. Ici les interventions du système sont impératives, les erreurs sont soulignées et demandent correction de la part de l'apprenant.
- *Assistance mixte* : la relation apprenant-assistance dans les modalités précédentes peut être imaginée comme suit : plus l'assistance est proactive plus l'apprenant est passif et inversement. Il s'agit de trouver le bon dosage en fonction des besoins de formation des apprenants. C'est par une coopération apprenant/système que l'apprentissage sera le mieux favorisé. Voilà pour quoi de plus

- en plus de systèmes adoptent un mode mixte de conseil où tantôt c'est l'apprenant qui consulte un assistant passif et, tantôt, c'est l'assistant proactif qui intervient pour donner un conseil ou suggérer un élément de solution.
- *Assistance collaborative* : sur un autre plan, on commence à prendre en compte des contextes de collaboration qui tiennent compte de la présence d'autres agents que l'apprenant et le système, notamment le formateur et les co-apprenants. De telles situations favorisent le partage des rôles de conseil et d'intervention pour tenir compte de la dimension coopérative de l'apprentissage. Dans un contexte comme celui de classe virtuelle, il est prévu que le système assistant sera surtout utilisé, de façon passive ou active, pour donner des conseils sur la démarche individuelle de l'apprenant, ainsi que sur ses interactions avec le groupe. Des interventions plus globales sont réservées au formateur, qui disposera d'un portrait des démarches individuelles et collectives fourni par le système d'assistance.

2.2.6 Assistance contextuelle

Une assistance contextuelle est obtenue à partir d'un point spécifique de l'état du logiciel appelé contexte, fournissant une aide pour la situation associée à cet état. Elle permet aux utilisateurs de recevoir de l'aide dans l'interface actuelle avec laquelle ils interagissent, plutôt que dans une autre interface d'aide.

[107] présente un outil de création d'aide contextuelle permettant aux utilisateurs d'appliquer des captures d'écran et d'écrire des scripts simples. L'aide contextuelle a été appliquée au guidage initial pour les nouvelles fonctionnalités dans le contexte d'une interface de carte [57], de tutoriels à base de stencil pour jeunes écoliers [58], de la programmation dans le contexte de l'éditeur Eclipse [11] et des tâches pratiques telles que la réservation de vols dans le contexte d'un navigateur Web [18]. L'importance des opinions et des avis des utilisateurs a mis en évidence la nécessité d'aider les utilisateurs à produire de meilleurs avis. [40] décrivent le développement et l'évaluation d'un assistant de commentaires sous la forme d'un plug-in de navigateur conçu pour fonctionner avec des sites majeurs comme Amazon. Cet assistant fournit aux utilisateurs des suggestions qui s'adaptent automatiquement au fur et à mesure que l'utilisateur écrit leurs commentaires.

Afin de fournir une assistance contextuelle, il est nécessaire de disposer au minimum d'informations sur l'outil, la tâche, les compétences et les préférences de l'utilisateur. Généralement, un modèle de contexte est défini pour décrire ces informations contextuelles et peut être instancié pour différentes situations. Des techniques de raisonnement sont ensuite utilisées pour déduire des connaissances pouvant être utiles au système d'assistance, comme c'est le cas dans les approches basées sur des ontologies ou celles basées sur les expériences passées. Dans ces dernières, le contexte est modélisé à l'aide d'une base de connaissances constituée de traces d'interaction.

2.3 Traces d'interaction

Généralement, une trace est définie comme l'influence d'un événement sur son environnement. Nous nous intéressons dans cette thèse à ce qui reste du passé, ce qui définit la trace comme *Une marque laissée par une action ou une activité*. Les traces sont liées à ce qui reste, à ce qui peut être observé dans ce processus ou ces actions. Elles permettent d'observer l'évolution des actions au sein de ce processus. [110]

Les traces sont la conséquence d'un ensemble d'éléments contextuels. Elles sont donc fortement liées au contexte dans lequel l'activité se produit. Par exemple, comme il a neigé, il est possible de voir les traces d'animaux dans la neige. Dans ce cas, le contexte météorologique est étroitement lié à la trace. Dans d'autres cas, les éléments contextuels sont nécessaires à l'interprétation des traces. Ainsi, le contexte doit être pris en compte lors de la collecte des traces afin de les interpréter correctement. [110]

Une trace numérique est une séquence d'éléments observés dans le temps, soit des interactions humaines médiées et inscrites dans et par l'environnement numérique lui-même de l'activité de l'utilisateur, soit une séquence d'actions et de réactions entre un humain et un ordinateur. Si nous utilisons la définition générale de la trace et que nous la spécialisons pour les traces numériques, elle devient : la trace numérique est faite à partir d'empreintes numériques laissées volontairement (ou non) par et dans l'environnement numérique lui-même pendant le processus numérique [28]. Les traces numériques fournissent des données sur ce qui a été effectué dans l'environnement numérique (par exemple, sur quoi vous avez cliqué, recherché, aimé, où vous êtes allé, votre emplacement, votre adresse IP, ce que vous avez dit, ce qui a été dit sur vous, etc.).

2.3.1 Définitions

Nous présentons dans cette section les définitions que nous adoptons et qui sont celles proposées par le laboratoire LIRIS sur le concept de trace, ainsi que les concepts afférents.

Trace, observé : une trace est composée d'éléments temporellement situés appelés obsels (pour observed element / élément observé). Ces derniers sont des éléments considérés comme potentiellement porteurs de sens dans l'activité tracée. Un obsel est constitué essentiellement : d'un type rattachant cet obsel à une catégorie explicite d'éléments observés, et d'un ensemble d'attributs de la forme $\langle \text{attribut} : \text{valeur} \rangle$.

m-trace : pour modeled trace ou trace modélisée, est une trace associée à un modèle qui en fournit un guide de construction et de manipulation. Un modèle de trace doit définir :

- la manière de représenter le temps,
- les types d'obsels permettant de décrire l'activité,
- pour chaque type d'obsel, les types d'attributs possibles,

— les types de *relations* binaires que peuvent entretenir les obsels entre eux.

Un Système de Gestion de Bases de Traces (SGBT) : est un outil informatique permettant la manipulation et la transformation de traces modélisées. En effet, un SGBT joue le même rôle qu'un SGBD (Système de Gestion de Bases de données) dans les applications standard, mais gère plutôt des traces modélisées (m-traces). Le SGBT est alimenté par un ensemble de collecteurs, dont le rôle est de récolter les informations nécessaires à la constitution des traces.

kTBS (kernel for Trace Based Systems) : est un système informatique mettant en pratique la notion de SGBT, développé au sein du laboratoire LIRIS. kTBS propose plusieurs fonctionnalités de gestion des traces, à savoir la création, l'interrogation, la transformation et la visualisation. kTBS utilise le formalisme RDF [91] pour décrire les traces et les modèles, et expose ses données dans différents formats (XML, JSON, Turtle).

2.3.2 Les traces dans la littérature

Le nombre et surtout la diversité des finalités attribuées aux traces d'interaction dans les approches en question sont un obstacle à toute tentative de classification claire. Pour n'en citer que quelques-uns : analyse d'utilisabilité, d'utilité ou de l'ergonomie des interfaces (IHM), modélisation utilisateur, caractérisation des comportements d'opérateurs (Psychologie Cognitive), caractérisation des parcours sur le Web (Sciences de l'information), assistance à l'utilisateur, personnalisation d'interface (Informatique), création et validation de scénarii pédagogiques, ou encore réflexivité des activités d'apprentissage (EIAH), etc. Peut-on distinguer les approches où les traces d'interaction sont exploitées en dehors de l'activité observée (dans une posture d'analyse), de celles où les traces le sont au sein même de ladite activité (dans une posture de réflexivité). Dans ce second cas, la trace de l'interaction est rendue disponible d'une façon ou d'une autre à l'utilisateur de l'environnement, qui est lui-même à l'origine de l'interaction en question. [62]

Si l'on trouve trace des traces d'interactions dans de nombreux domaines, certains constituent des pôles de recherche particulièrement actifs. C'est le cas des recherches liées au Web tant en termes d'analyse que de réflexivité. Du côté de l'analyse d'abord, on retrouve par exemple de manière commune chez tous les hébergeurs de sites (ou de blogs) des fonctionnalités de traçage de l'activité des visiteurs à l'image de ce que fournit Google Analytics. Mais on trouve également sur le sujet des travaux exploratoires visant à caractériser les parcours des internautes sur le Web [10], et plus généralement les travaux dits de Web Mining ou Web Usage Mining [22], qui se basent sur des analyses statistiques des logs-files. Du côté de la réflexivité cette fois, on trouve de plus en plus de fonctionnalités additionnelles de navigateurs (Firefox le plus souvent) permettant à un internaute de tracer sa navigation un peu à l'image de Google Web History. Des recherches plus poussées encore sont également menées avec des technologies similaires, à l'image du NavTracer, ou en y ajoutant des couches logicielles supplémen-

taires afin de fournir des assistants à la navigation, c'est le cas Letizia [65], ou d'étudier la visualisation de parcours ou de données temporelles [68].

Ces différents travaux exploitent le fait que des traces d'interactions « existent déjà », si l'on peut dire, à travers le fonctionnement des navigateurs. Dans le domaine des EIAH (Environnements Informatiques d'Apprentissage Humain), autre pôle très actif, on retrouve pareillement des démarches d'analyse des interactions [86], mais également (et de manière plus sophistiquée) des tentatives d'assistance à l'apprenant s'appuyant sur ses propres traces d'interaction [80]. Le trait particulier de certains de ces travaux est de ne pas se baser uniquement sur des logs existants comme précédemment, mais de parfois créer de toutes pièces un environnement dans lequel les interactions ont été déterminées à l'avance [98] délimitant ainsi de fait le champ d'observation pour les traces.

2.3.3 Modélisation des traces

Les traces numériques doivent être accessibles à l'homme sous une forme intelligible. A cette fin, les calculs interprétatifs menés sur les traces pour les transformer doivent être intelligibles et modifiables par l'homme. Il est évident que quel que soit l'observateur, il est nécessaire que les traces soient intelligibles pour qu'elles aient le statut d'inscription de connaissance. La reformulation selon telle ou telle interprétation aboutissant à de nouvelles traces doit être également accessible et intelligible.

La dynamique de l'expérience, constitutive du temps doit être intrinsèquement présente dans la trace numérique. On touche ici à une spécification particulière de ce type d'inscription de connaissances. A chaque trace numérique doit être associé un dispositif permettant d'en constituer la temporalité. L'objet trace informatique proposé pour tenter de satisfaire ces hypothèses est ce que nous appelons une trace modélisée. Une trace modélisée est constituée d'une partie trace (séquence d'observés temporellement situés) et d'une partie modèle de trace (vocabulaire et contraintes sur les observés de la trace).

Une première formalisation de cette notion de trace modélisée a été réalisée dans le cadre de la thèse de Lotfi Settouti [95]. Une autre définition a été donnée au concept de trace modélisées et considère qu'une trace modélisée (M-Trace) est une séquence temporelle d'observés d'une activité interactive avec un environnement Informatique munie d'un modèle de trace dans un domaine temporel donné.

Un modèle de trace permet donc de « parler », d'établir un « discours » à propos des observés d'une trace comme instances d'un ensemble fini de classes d'objets. Chaque observé est temporellement situé conformément au domaine temporel de la trace (séquencement simple, datation en instants, datation par intervalles de temps, etc.).

2.3.4 Système à base de traces

La collecte des traces peut offrir différents services selon son destinataire ; ainsi, dans [96], quatre utilisations principales des traces d'interaction ont été identifiées : (1) prendre conscience de l'activité par l'apprenant et l'enseignant (awareness), (2) refléter l'activité de l'apprenant en lui offrant la possibilité de visualiser son activité (mirroring), (3) utiliser les traces pour guider l'apprenant dans son activité d'apprentissage en lui proposant par exemple l'action suivante à réaliser, (4) assister les analystes chercheurs en leur proposant des traces qui présentent généralement un niveau d'abstraction élevé permettant une analyse de la situation d'apprentissage en fonction des objectifs et hypothèses de recherche. Dans [95] un Système à Base de Traces modélisées (SBT) est présenté comme une sorte de Système à Base de Connaissances, dont la source de connaissance est l'ensemble des traces d'interaction d'un utilisateur avec un système. Un SBT manipule des traces modélisées. Une trace modélisée est une trace munie d'un modèle exprimant la sémantique de son contenu.

Un SBT est défini dans [96] comme : « tout système informatique dont le fonctionnement implique à des degrés divers la gestion, la transformation et la visualisation de traces modélisées explicitement en tant que telles ». Dans [95], les concepts de base d'un SBT ont été définis formellement. Il s'agit des notions de modèle de trace, trace modélisée, schéma (pattern), requête et transformation. Un modèle de trace définit un vocabulaire pour décrire une trace. Il permet de (1) préciser la représentation du temps dans la trace, (2) classifier les éléments observés ainsi que les attributs qui les décrivent, et les éventuelles relations hiérarchiques entre ces classes d'observés (classe au sens du paradigme orienté objet), (3) définir les types de relations pouvant exister entre les observés, et les éventuelles relations hiérarchiques entre ces relations. Un pattern permet d'exprimer un ensemble de critères à satisfaire par les éléments observés d'une trace modélisée. Il est possible d'exécuter des requêtes sur les traces en utilisant la notion de pattern. Une requête est donc définie par un nom, un pattern qui définit les filtres à exécuter et l'ensemble de variables à retourner.

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

La figure suivante présente l'architecture d'un système à base de traces modélisées. Un système de traçage commence par collecter les traces d'interaction. Le système de traçage construit alors des traces modélisées dites primaires souvent d'un niveau d'abstraction bas. Ces traces peuvent être capturées en temps réel à l'aide de sources de traçage actives et stockées dans un entrepôt actif, ce sont des M-Traces en ligne. Elles

peuvent sinon être des M-Traces collectées en différé et être stockées dans un entrepôt persistant. Le système de transformation exécute des transformations sur les traces en appliquant des filtres, réécrivant ou agrégeant des éléments de traces. Ceci peut donner lieu à des M-Traces plus pertinentes pour une utilisation plus spécifique. Le système d'interrogation permet d'interroger la base de traces modélisées pour extraire des informations spécifiques nécessaires à une étude donnée.

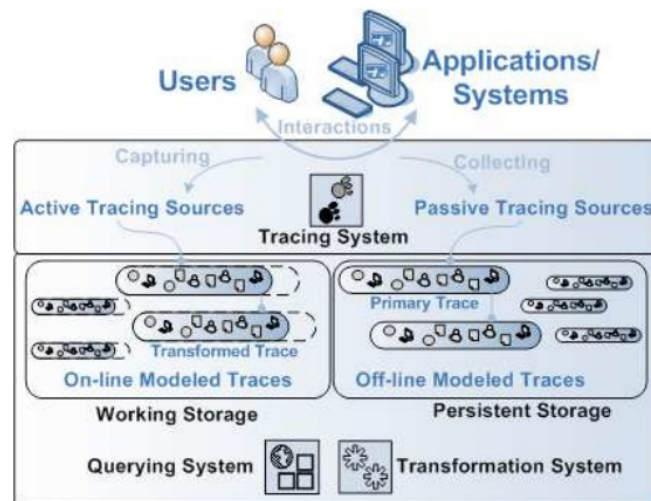


FIGURE 2.2 – Architecture d'un Système à Base de Trace [95]

Le SBT tel qu'il est formalisé est générique et peut être utilisé pour la gestion et la manipulation de traces d'interaction dans n'importe quel domaine d'application, et non pas seulement dans le domaine des EIAH. Ce cadre conceptuel a permis l'implémentation de systèmes dans des contextes spécifiques en proposant des modèles de traces et des transformations adéquats. SBT-IM [39] et TREAM [94], sont deux exemples d'application de ce cadre conceptuel dans le domaine de l'apprentissage collaboratif et de l'apprentissage individuel respectivement. ABSTRACT [43] est une autre application de SBT et propose un outil d'ingénierie des connaissances à partir des traces d'activité, destiné aux ergonomistes, et dédié à l'analyse et la modélisation de l'activité dans le domaine de la conduite automobile. Il semble cependant difficile d'implémenter un système générique qui soit utilisable dans n'importe quel domaine et contexte, du fait des différences au niveau des traces manipulées dans les différents domaines. Le cadre conceptuel des SBT définit formellement un méta-modèle générique permettant la modélisation d'une trace mais ne propose aucun modèle ou format concret pour décrire le contenu d'une trace. Ce travail reste dans un niveau théorique et doit être implémenté selon les besoins spécifiques de chaque système.

2.3.5 La collecte dans les systèmes de bases de traces

Un processus de collecte (voir figure suivante) est nécessaire pour construire ce que nous appelons les traces premières. Une trace première est une trace modélisée selon le premier modèle de collecte. Choisir les observés à conserver dans la trace première est naturellement une question ouverte. Dans les usages actuels des traces pour les EIAH, les situations de collecte sont contrastées : du « on prend ce qu'on a dans les logs » au « on instrumente soigneusement l'environnement pour récupérer les observés contrôlés et utiles », en passant par une instrumentation « attrape tout » comme un key logger par exemple. L'ingénierie des traces modélisées en est à ses débuts, mais comme il s'agit avant tout de traces d'interactions, il n'est pas impossible d'imaginer des techniques de génie logiciel permettant de paramétrer le modèle de collecte à partir des interfaces de programmation d'applications et dans ce cas, ce paramétrage pourrait être par exemple une des fonctions du SBT.

Le principe retenu pour le processus de collecte est de considérer des sources de traçage qui peuvent très variées comme l'illustre la figure suivante : les applications utilisées naturellement, mais aussi des textes d'annotation, des informations issues de vidéos, de bandes sonores et d'une manière générale de tout dispositif permettant de fournir des observés temporellement situé.

Dès l'instant où une trace modélisée première est ajoutée dans le SBT, sa collecte démarre et son stockage est assuré dans la base de traces. Il ne faut pas confondre collecte avec capture. La capture a lieu à l'origine de la production des observés ; c'est la collecte qui détermine le sous-ensemble d'observés retenus pour une trace modélisée donnée. Chaque trace modélisée est constituée de la séquence des observés s'instanciant dans le domaine temporel choisi. Cette trace première peut être visualisée soit par l'interface par défaut du SBT, soit par toute application spécialisée connectée au SBT.

Chaque trace est gérée par un identificateur unique et caractérisée par son propriétaire, avec des droits et protections spécifiques. Toute trace modélisée peut être fournie à une application pour réaliser des calculs ou l'exploiter en dehors du système de transformation du SBT, par exemple pour l'élaboration de profils d'apprenants, d'indicateurs de personnalisation, etc.

2.4 Assistance à base de traces

Un des avantages du développement d'un système basé sur les traces d'interaction est de faciliter l'analyse et la modélisation des activités. Suivant ces principes, plusieurs études ont implémenté des applications réelles basées sur les traces. [44] ont mis en place un système basé sur les traces pour modéliser l'activité de conduite à partir de traces recueillies avec un véhicule instrumenté. [70] a appliqué un raisonnement à base de trace (TBR) pour un Stream Mining. Le Stream Mining est l'extraction de connais-

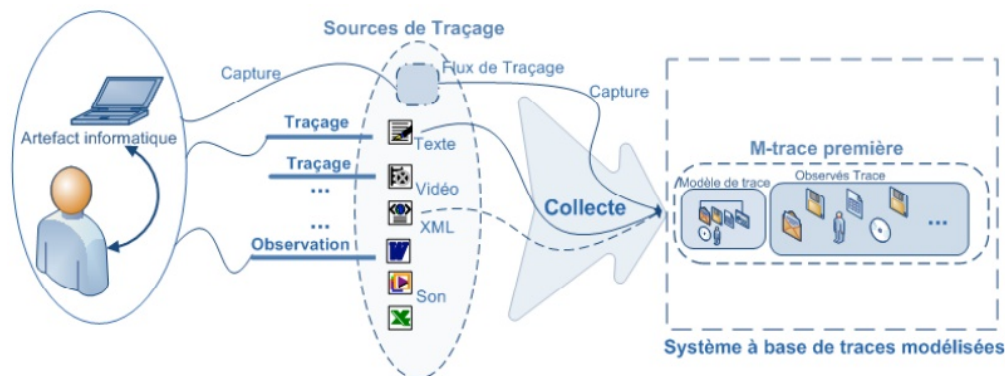


FIGURE 2.3 – Processus de collecte [95]

sances à partir d'enregistrements continus. Le TBR s'est avéré efficace pour aider les utilisateurs. [34] ont proposé un processus en trois étapes qui réutilise les traces pour une assistance contextuelle : collecte des traces à partir de capteurs, extraction des connaissances et assistance. Leur approche réutilise l'expérience personnelle de l'utilisateur comme alternative aux systèmes traditionnels avec un modèle de contexte explicite associé à des capacités de raisonnement sur ce modèle. Ginon et al. [45] ont proposé un modèle d'assistance générique pour la spécification et la mise en œuvre de systèmes d'assistance personnalisés pour une application cible existante, sans avoir à la modifier. Dans le cadre du projet Kolflow [29], les auteurs ont créé un outil de support pouvant aider les utilisateurs à mieux gérer la fusion de ressources dans un contexte distribué. En particulier, ils se concentrent sur l'activité collaborative de développement des connaissances, soutenue par un réseau de wikis sémantiques distribués.

Le TBR a été spécifiquement étudiée dans le cas des systèmes d'apprentissage humain. Les systèmes de tutorat intelligents et les outils d'apprentissage collaboratifs sont, par nature, conçus pour aider les apprenants. Ces outils utilisent souvent différentes formes de traces comme entrées pour l'aide [28]. Dans ces outils, les enregistrements d'expériences passées, ainsi que des explications, jouent un rôle important. L'application Visu fournit un bon exemple d'utilisation de traces dans un espace d'apprentissage collaboratif [17]. Un autre exemple est celui de [93], un système d'aide adaptatif basé sur des traces d'interaction permettant aux tuteurs et aux apprenants de s'aider mutuellement en partageant des traces.

Notre cas d'utilisation de traces pour offrir un contrôle pédagogique dans un contexte d'EIAH est en réalité un autre exemple de ces systèmes exploitant les traces. La première solution que nous proposons au problème de contrôle pédagogique en EIAH est d'offrir une panoplie d'outils (automatique et non automatique) aux systèmes d'apprentissage et aux analystes leur permettant d'intervenir avec la façon la plus appropriée. Nous entendons par outils celles d'analyse, de transformation, de requêtage et de visualisation proposée par les Systèmes à Base de Traces (SBT). Le travail

sur la collecte de traces variées des EIAH est une étape inévitable dans le processus d'utilisation des SBT et celle que nous aborderons dans l'une de nos investigations. La deuxième solution propose en dehors des SBT mais toujours en exploitant les traces, de les interpréter et de les faciliter pour soutenir l'analyste à choisir l'assistance (instantanément ou avec des artefacts de configuration) la plus appropriée. L'interprétation des traces est le sujet d'une autre investigation que nous avons menée, elle est basée comme nous allons le voir dans le chapitre 5 sur la réécriture de la trace en patterns (de haut niveau sémantique) en appliquant un appariement et une mesure de similarité. La deuxième solution est aussi un autre exemple sur l'utilisation des traces d'interaction pour faciliter le contrôle et l'assistance des apprenants dans un contexte d'EIAH via leur interprétation et réécriture.

2.5 Conclusion

A travers ce chapitre on a pu explorer et faire un aperçu sur les questions afférents à l'assistance et aux traces d'interaction après plusieurs consultations et discussions de systèmes existants et après une recherche bibliographique approfondie. Ceci va bien évidemment éclaircir et expliquer les concepts techniques clés que nous allons rencontrer ou manipuler le long de ce mémoire et mettre en valeur nos investigations. En outre, connaître l'état actuel du domaine va nous permettre de bien spécifier notre travail, le différencier et le valoriser parmi d'autres.

3

Les traces dans les systèmes existants

Plan du chapitre

1.1	Contexte	2
1.1.1	Définitions	3
1.2	Question de recherche	4
1.3	Organisation de la thèse	7

3.1 Introduction

Les traces numériques sont des inscriptions temporalisées liées à un système informatique donné et ce pour diverses exploitations. Bien que la naissance des traces date de plusieurs dizaines d'années, elles ont subi une mutation importante cette dernière décennie. En effet, de nombreux systèmes informatiques ont commencé à produire et/ou exploiter les traces, mais chacun avec son propre format, son propre contenu et pour ses propres objectifs. Le concept de trace s'est bien développé, par rapport aux formats possibles adoptés, aux contenus considérés et aux exploitations visées. Cependant, des défis majeurs se posent : celui de leur compréhension et celui de leur interopérabilité. Le problème de compréhension se pose généralement durant les processus d'exploitation des traces ayant pour but la visualisation, l'analyse et la déduction de connaissances. Notre travail s'intègre dans le domaine de l'intégration de données, pour justement proposer une solution au problème de l'interopérabilité. Nous visons à intégrer des traces variées dans un formalisme unique pour permettre ensuite leur exploitation.

Un système tracé est un système capable de générer des traces numériques d'interaction liées aux activités qu'il propose à l'utilisateur. Les systèmes tracés actuels utilisent des traces numériques très variées, à des fins aussi variées. Après consultation et analyse des traces d'un ensemble de systèmes tracés, nous avons abouti à quelques constatations qui nous ont été utiles pour proposer notre approche de collecte sujet du chapitre suivant.

Ce chapitre peut ainsi être considéré comme un état de l'art sur les traces d'interaction et leurs systèmes générateurs. Plus précisément, nous présenterons dans la section suivante une sorte de résumé sur le domaine des EIAHs, leurs architecture et fonctions, leur catégories ainsi que les problématiques de recherche en relation. La section d'après présente une étude d'une collection de systèmes produisant des traces et nous mettrons l'accent sur le contexte et les fonctions essentielles de chacun d'eux. L'étude bien évidemment contient une partie pour chaque logiciel expliquant le format l'organisation et le contenu des traces produites. Avant de commencer l'étude on a préféré expliquer les objectifs et les besoins qui nous ont poussé à la faire. La section d'après contient un récapitulatif de cette étude et pour quelques points une confrontation afin de bien montrer les variétés. Nous terminerons ce chapitre par une conclusion résumant tout le chapitre.

3.2 Quelques repères pour les EIAHs

La recherche sur les Environnements Informatiques d'Apprentissage Humain (EIAH) a produit des Logiciels Educatifs Intelligents et Interactifs pour l'enseignant et l'apprenant. Le terme EIAH « Environnement Informatique pour l'Apprentissage

Humain » est l'héritier d'une succession d'autres termes qu'a connaît le domaine d'apprentissage par l'outil informatique. De l'enseignement programmé et de l'EAO « Enseignement Assisté par Ordinateur » à l'EIAO « Enseignement Intelligemment Assisté par Ordinateur » avec la propagation des techniques d'intelligence artificielle puis à l'« Enseignement Interactif d'Apprentissage par Ordinateur » où l'accent est sur l'importance de l'interactivité dans ces systèmes. Le terme EIAH dénote une évolution vers le couplage entre l'homme et la machine, notamment à travers les Technologies de l'Information et de la Communication (TIC) et élargit le champ d'étude à l'apprentissage humain dans toutes ses déclinaisons (enseignement, formation, autodidaxie, diffusion de connaissances, etc.) cite tchounikine2002quelques. Récemment, le terme EIAH est communément utilisé pour désigner tout environnement informatique conçu pour favoriser un apprentissage humain.

Le domaine des EIAHs couvre une diversité de travaux et de systèmes. Leur point commun est la mise en relation d'une intention didactique et d'un environnement informatique. Le domaine du web par exemple, a connu une des forme de ces systèmes à travers les hypermédias éducatifs et les plateformes d'apprentissage. Nous pouvons citer dans ce contexte les hypermédias adaptatifs et intelligents, les plateformes EMS, les plateformes d'e-Learning et les MOOC, qui ont tous utilisé le web comme support technologique. Un peu plus avant ce champs a aussi connu la mode des systèmes tuteurs intelligents qui ont été intégré dans grand nombre de systèmes informatiques d'apprentissage. Les simulateurs éducatif ou pédagogiques et les micro monde est une autre catégorie où l'accent est mis sur la pratique et sur la mise en situation proche du réel. Pas loin de cette idée d'apprentissage indirect et souple dans les simulateur, d'autres variantes plus ludiques ont vu le jour, les jeux éducatifs ou les « Serious Games ». L'idée derrière est de proposer des contextes plus agréables orientés non exclusivement à de jeunes apprenants. Ces systèmes profitent du plaisir attient par les joueurs pour leur transmettre implicitement des éléments de connaissance. En effet, les exemples de formes d'EIAH que nous avons présentés dans ce paragraphe ne sont qu'un sous ensemble des systèmes et travaux faits dans le domaine depuis des décennies d'années et la liste est bien sur plus longue.

Ces logiciels, implantent des fonctionnalités variées. A la base ils sont conçus pour présenter des contenus pédagogiques aux apprenants à travers des interfaces interactives éventuellement adaptées. Selon le domaine, les EIAHs peuvent proposer des activités variées aux apprenants à savoir, des exercices d'entraînement de différents types, des travaux pratiques, des animations et des simulations éventuellement interactives à suivre, des activités collaboratifs, des tests de passage a but d'évaluation, etc. Pour pouvoir assurer toutes ces fonctionnalités, les EIAHs intègrent plusieurs modules. Ce fait explique l'aspect complexe de ces systèmes informatiques et les spécifie relativement aux systèmes d'information ou même aux systèmes à base de connaissance par exemple. Les EIAHs intègrent généralement un modèle de domaine décrivant le domaine à enseigné. Ce dernier va servir comme référence pendant les processus d'apprentissage et d'évaluation. Ils intègrent aussi un modèle apprenant formé d'un en-

semble de variables utilisé à des fins de profilage et de caractérisation. Un modèle apprenant est en effet, derrière les tâches de personnalisation des apprentissages et l'adaptation des interfaces. L'aspect interactif et visuel est central dans les EIAHs ce qui explique la nécessité d'avoir un module d'interface ou de présentation dans ces systèmes. Ces systèmes peuvent aussi intégrer si besoin : un module de communication dans un contexte collaboratif, un générateur d'exercice, un module d'évaluation, un module de traçage ainsi que d'autres modules plus spécifiques.

Plusieurs questions et thématiques de recherche sont soulevées autour de ce domaine ce qui a conduit à des systèmes de plus en plus utiles et efficaces. Parmi ces thématiques celle de l'ingénierie des EIAHs qui s'intéresse aux techniques de développement de ces logiciels tout en considérant ses spécificités didactiques. En effet, plusieurs questions sont posées et traitées autour des modules composant les EIAHs. Nous pouvons citer les ateliers s'intéressant à :

- la modélisation de l'apprenant,
- la représentation et la description du domaine enseigné avec des outils variés,
- la génération automatique du contenu (exercices, interfaces, ...)
- l'évaluation automatique/semi automatique des apprenants
- l'adaptation des interfaces et des contenus
- l'intégration de l'intelligence pour imiter le comportement humain
- l'assistance des apprenants dans des contextes variés grâce à de nombreuses techniques
- ...

Pareillement, le travail que nous présentons dans ce manuscrit n'est qu'un exemple et une participation dans ces efforts afin de donner un plus dans le champ de l'assistance à l'apprenant dans ces logiciels.

3.3 Etude et analyse des traces existantes

3.3.1 Objectif

Au début du travail dans cette thèse, l'étude des traces et des systèmes tracés n'était prévu et fixée comme une étape à franchir. Néanmoins, le temps où nous avons choisi d'utiliser les possibilités d'analyse et de requête des SBT pour assurer l'assistance pédagogique cherchée, nous étions forcés à étudier ces traces avant de passer à la collecte. En effet, connaître l'existant des traces est une étape inévitable pour réussir le processus de collecte. L'absence de standard pour les traces fait que chaque système tracé choisit le type, le format et le contenu qu'il arrange. Nous pensons qu'une étude pas forcément épuisante des traces existantes va orienter et optimiser nos efforts. Puisque une telle étude va nous informer sur les contenus et les formats à gérer cela adaptera bien notre solution et nous conduit à choisir les moyens les plus adéquats. D'autre part, connaître

la large variété des traces va nous amener à limiter notre champ d'étude autrement dit le type des traces et systèmes à traiter avec la façon la plus satisfaisante.

3.3.2 Traces étudiées

Dû à leur intérêt les traces d'interaction sont de plus en plus utilisées. Les serveurs réseaux traitaient les log, les environnements d'apprentissage a petits ou grand public, les navigateurs web ne sont plus les seuls à en profiter la liste est encore ouverte. Parmi toute cette liste ce sont les logiciels d'apprentissage qui nous intéressent le plus et aucune considération n'a été prise exceptant cela. La seule considération que nous pouvons peut être ajouter est la possibilité d'accès aux traces des systèmes tracés. Notre choix des systèmes dont nous allons étudier les traces s'est référé au travail de thèse de Marie Levevre [63]. Lefevre présente une liste de logiciels pédagogiques produisant des traces d'interaction. Bien que la majorité des logiciels présentés sont des EIAH, quelques outils d'analyse et de diagnostique font aussi partie de la liste. D'autres systèmes connus dans notre contexte sont ajoutés à la liste pour des raisons de diversité.

3.3.2.1 DataShop

DataShop [79] est un référentiel de données et une application Web pour les chercheurs en sciences d'apprentissage et en Data Mining éducatif. Il fournit un stockage de données sécurisé ainsi qu'un ensemble d'outils d'analyse et de visualisation disponibles via une interface Web. Principalement, DataShop stocke les interactions des apprenants à partir de supports de cours en ligne comprenant des tuteurs intelligents. Les données proviennent des sept cours du PSLC (Pittsburgh Science of Learning Center) : algèbre, chimie, chinois, anglais, français, géométrie, physique ainsi que d'autres sources externes. L'application Web fournit plusieurs outils pour faciliter l'analyse et la visualisation des données du référentiel. Ces outils peuvent être utilisés conjointement pour lancer une analyse des données : il est possible de déterminer si les étudiants apprennent en visualisant les courbes d'apprentissage (via Learning Curve), puis d'examiner les problèmes individuels(via Error Report), les composants de connaissances (via Dataset Info) et les étudiants afin d'analyser les mesures de performance(via Performance Profiler).[60]

Les données peuvent être importées dans le référentiel DataShop via XML ou un format de fichier texte délimité par des tabulations. Dans l'autre sens, DataShop propose diverses options d'exportation de données via son application Web, chacune étant livrée dans un fichier texte délimité par des tabulations [60]. Les traces produites par DataShop sont donc des fichiers texte organisés en plusieurs colonnes d'informations délimitées par des tabulations et un caractère fin de ligne. La première ligne est réservée aux intitulés des colonnes.

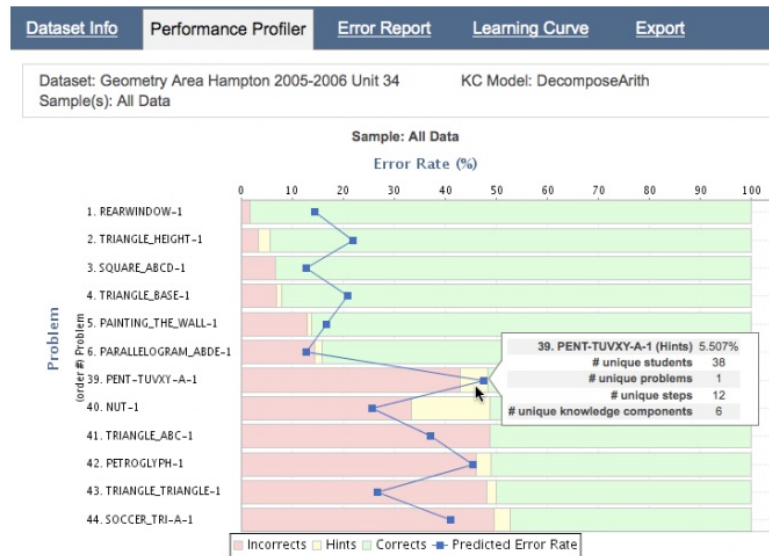


FIGURE 3.1 – L’outil Performance Profiler de l’application Web DataShop

3.3.2.2 TELEOS

L’environnement TELEOS présente un logiciel de simulation 3D fournissant aux stagiaires des exercices de simulation de vertébroplastie et de fixation par vis sacro-iliaque. Une base de données de cas cliniques fournit aux apprenants des modèles de patients qu’ils peuvent librement en choisir pour pratiquer. Ces derniers sont des dossiers d’informations complètes sur les patients, rassemblés au Service d’Orthopédie et de Traumatologie du CHU de Grenoble ; puis modélisés et sauvegardés dans une base de données dédiée. Au cours d’une séance de simulation, le logiciel permet aux apprenants de visualiser les modèles des patients et les instruments chirurgicaux en 3D. La plateforme intègre un fluoroscope utilisé pour la radiographie facilitant le guidage visuel, un bras haptique qui enregistre la reproduction valide des gestes chirurgicaux et d’un oculomètre traçant les points fixés par l’apprenant tout au long d’une intervention. Les apprenants peuvent pratiquer des gestes chirurgicaux et des activités impliquées dans la chirurgie orthopédique percutanée du monde réel comme : marquer des zones ciblées sur la peau du patient, faire fonctionner le fluoroscope, générer et visualiser des rayon-x ; utiliser le trocart pour l’injection de ciment ou l’implant à vis. TELEOS comprend une base de données de cours théoriques, un outil auteur pour l’administration du modèle de connaissances, un module de diagnostic des connaissances et un module de rétroaction didactique.[101] [100]

Les traces de TELEOS viennent de trois sources différentes du système le logiciel simulateur, le bras haptique et l’oculomètre d’où leur hétérogénéité [101]. Elles contiennent en général des informations parfois très techniques organisées en colonnes

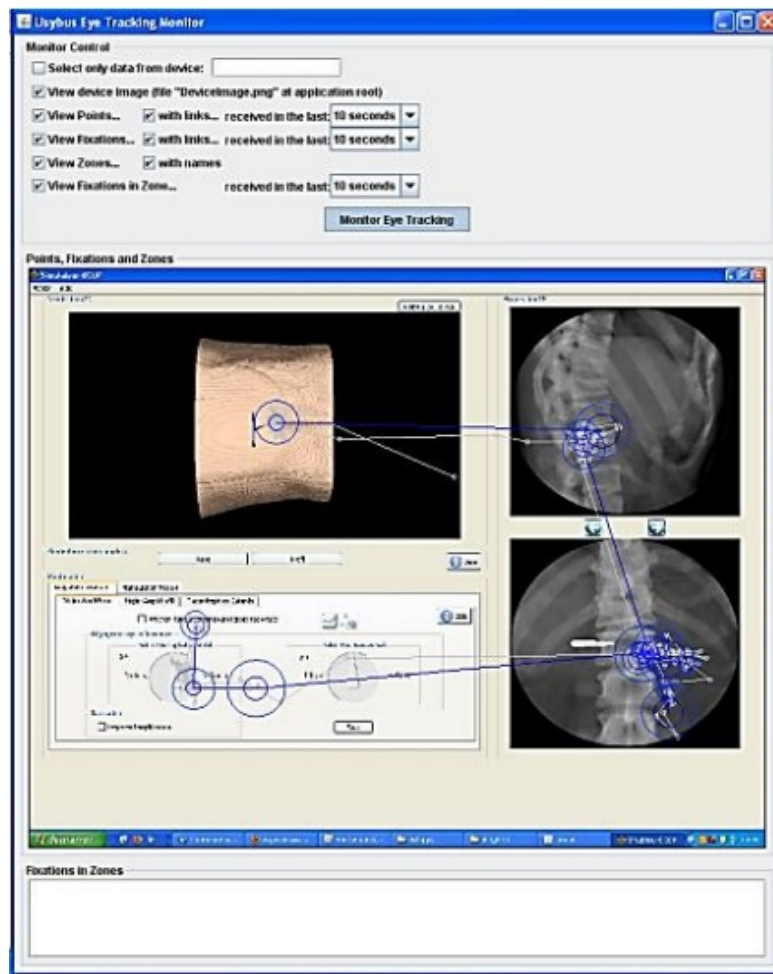


FIGURE 3.2 – Interface du simulateur TELEOS

éventuellement chevauchées avec un séparateur ";" et un caractère fin de ligne. La première ligne est réservée aux intitulés des colonnes.

3.3.2.3 APLUSIX

APLUSIX [27] est un environnement d'apprentissage pour l'algèbre formelle. Ce système comprend un éditeur avancé d'expressions algébriques qui affiche les expressions sous la forme habituelle et permet de modifier les expressions sous cette forme. Cet éditeur est basé sur la structure des expressions algébriques pour les fonctions supérieures (sélection, couper, copier, coller, glisser-déposer), par exemple, seules les sous-expressions algébriques peuvent être sélectionnées. En conséquence, les élèves font leurs propres calculs, comme ils le font avec papier crayon. Ceci est très différent des autres environnements d'apprentissage pour l'algèbre formelle qui nécessitent

l'utilisation d'une commande pour effectuer chaque action. Outre ce mode d'interaction intense, la plupart de ces systèmes effectuent tous les calculs. Dans ces environnements basés sur des commandes, l'étudiant ne peut pas faire d'erreur et ne peut pas apprendre de la correction de ses erreurs.

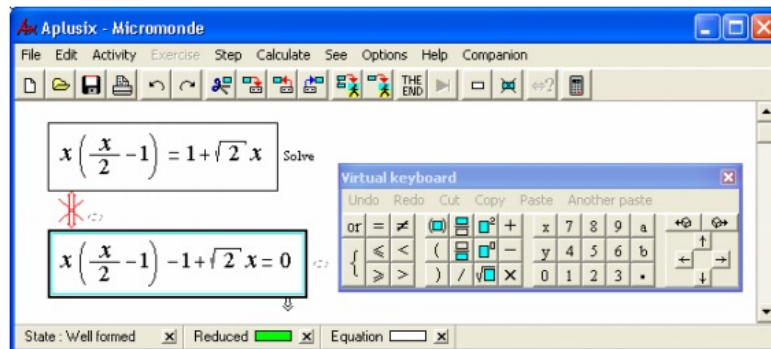


FIGURE 3.3 – Interface du système APLUSIX

Les traces d'APLUSIX [26] se présentent comme des fichiers texte en format .csv. Les traces (fichiers) peuvent comporter plusieurs sessions d'utilisation d'APLUSIX. Chaque session est divisée en deux parties, un en-tête contenant le contexte d'utilisation avec des informations sur l'élève, les paramètres et les options utilisés dans l'outil et une partie pour les actions enregistrées pour l'apprenant. Chaque ligne action contient un ensemble d'informations séparées par des ";" correspondantes à cette action et se termine par une fin de ligne. Les noms des informations entête précèdent leurs valeurs et les noms des informations de l'action sont placées au dessus de leurs valeurs.

3.3.2.4 EDDBA

EDDBA (Exercices DataBase about Algorithms) [19] est une application web pour l'apprentissage de l'algorithmique réalisée par Denis Bouhineau (Université de Grenoble). EDDBA est fondé sur la technologie AJAX, il utilise ainsi entre autres, une partie cliente (Javascript+HTML/CSS), une minuscule partie PHP fait le lien avec une troisième partie garantissant la persistance des données d'EDDBA (en BD, via un serveur MySQL). L'application permet à un apprenant de s'exercer en algorithmique en commençant par le choix d'un exercice dans une liste issue d'une BD selon plusieurs critères. L'apprenant peut ensuite rédiger sa solution et choisir un langage de programmation parmi plusieurs choix. EDDBA pourrait ainsi tester cette solution par exécution sur des jeux d'essai de référence et de faire la comparaison des résultats obtenus avec les résultats de référence (stockés dans la BD).

Les traces EDDBA se présentent comme deux types de fichiers textes, un centré sur l'activité de l'apprenant l'autre sur les exercices. Le nombre et le type des informations des actions faites par les utilisateurs dépendent du type de l'action. Ces informations



sont déclarées sans entête avec un séparateur " : " et un caractère fin de ligne. D'une façon non structurée la partie exercice contient : les codes successifs écrits par les apprenants pour cet exercice, les résultats des tests effectués sur ces algorithmes et un tableau récapitulatif.

3.3.2.5 Andes

Andes [92] est un environnement d'apprentissage pour la physique classique mis au point depuis 1996 par des chercheurs du Learning Research and Development Center de l'Université de Pittsburgh et de la United States Naval Academy. Ce tuteur intelligent permet aux étudiants de résoudre des problèmes de physique dans un environnement qui fournit une visualisation, une rétroaction immédiate et une aide procédurale et conceptuelle. Andes se compose d'un module auteur et d'un module étudiant. Il enseigne actuellement aux étudiants les domaines des forces statiques, de la cinématique translationnelle et rotationnelle, de la dynamique translationnelle et rotationnelle, de l'énergie et du moment linéaire et angulaire. Dans un scénario typique d'apprentissage d'Andes, les élèves lisent le problème, dessinent des vecteurs et des axes de coordonnées, définissent des variables et saisissent des équations. [103]

Andes produit des traces sous forme d'un seul fichier texte contenant après une ligne d'entête, une colonne de temps et une autre pour diverses informations tracées. Ces dernières sont variées en genre et en nombre tel que chaque ligne contient une liste composée des noms d'informations suivies des valeurs de ces informations. Les caractères "blanc" et le "l" sont utilisés pour la séparation avec un caractère de fin de ligne.

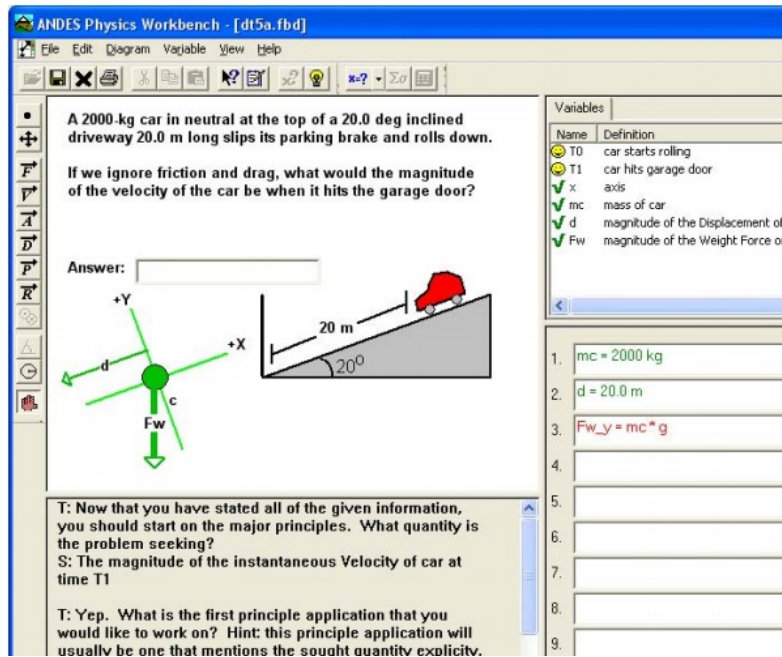


FIGURE 3.5 – Interface d'Andes

3.3.2.6 SQL-Tutor

SQL-Tutor [76] est un système tuteur intelligent qui fournit un environnement permettant aux apprenants de pratiquer et de développer leurs compétences en écriture de requêtes SQL SELECT. Le tuteur suppose que les apprenants ont couvert les concepts de bases de données relationnelles et de requêtes SELECT avant de l'utiliser. Il est conçu comme un outil pratique qui complète l'apprentissage des concepts associés via d'autres sources. SQL-Tutor est un système tuteur intelligent qui fournit un environnement permettant aux apprenants de pratiquer et de développer leurs compétences en écriture de requêtes SQL SELECT. Le tuteur suppose que les apprenants ont couvert les concepts de bases de données relationnelles et de requêtes SELECT avant de l'utiliser. Il est conçu comme un outil pratique qui complète l'apprentissage des concepts associés via d'autres sources. SQL-Tutor intègre un peu moins de 300 problèmes au total, qui sont présentés dans 13 bases de données différentes. La modélisation basée sur les contraintes (CBM) est au cœur de SQL-Tutor. Avec la CBM, la connaissance du domaine est contenue dans des contraintes contre des solutions correctes, chaque contrainte représentant «un fait ou un principe atomique du domaine». Il existe plus de 700 contraintes dans SQL-Tutor. Pendant que l'apprenant travaille sur un problème donné, SQL-Tutor fournit des aides via l'un des six niveaux de feedback du domaine, allant du Simple Feedback à la Solution Complete. SQL-Tutor a été développé pour fournir aux participants un retour adaptatif sur la motivation et la métacognition, basé sur les rapports d'auto-efficacité des participants et sur les problèmes qu'ils avaient

déjà résolus. [52]



FIGURE 3.6 – Interface de SQL-Tutor

SQL-Tutor produit comme trace, un seul fichier texte séparé en 3 parties : la première contient des informations du contexte, la deuxième la succession des solutions de l'apprenant et la troisième les listes de contraintes correctes et violées dans les solutions. Dans toutes ces parties, les informations sont organisées chacune dans une ligne sous forme de couple : nom de l'information, valeur de l'information. Les séparateurs utilisés sont les ":" entre les couples et les caractères fins de lignes.

3.3.2.7 TP-Elec

TP-Elec [66] est un micro-monde pour les travaux pratique en électricité réalisé par l'équipe METAH du laboratoire LIg de Grenoble en collaboration avec des enseignants de physique. Il permet la manipulation directe des principaux composants électriques utilisés dans les travaux pratiques d'électricité : le générateur à tension variable, les piles électriques, les résistances, les lampes, les appareils de mesure (voltmètre et ampèremètre), le fusible, l'interrupteur, la diode électroluminescente. . Dans TP-Elec l'apprenant peut : sélectionner et poser les composants électrique sur le plan de travail, les relier et les alimenter. Le logiciel TPElec est une applet Flash pouvant être très facilement intégrée à une plate-forme web. [75]

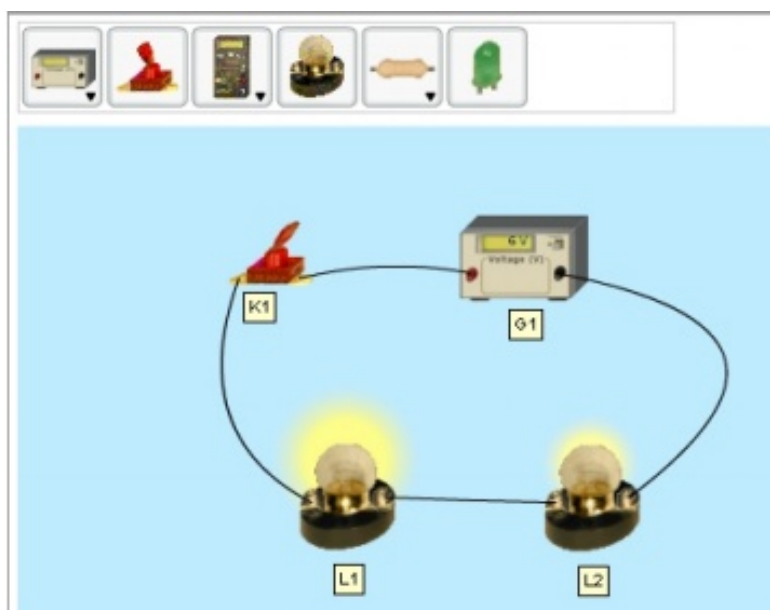


FIGURE 3.7 – Interface du micromonde TPElec

TP-Elec produit comme trace un seul fichier XML contenant la liste des circuits réalisés par l'élève, chaque circuit est composé du triplet : ListeComposants (composants électriques manipulés), ListeVuePhoto (correspond aux vues des circuits réalisés avec les positions sur l'écran), des actions de l'apprenant sur ces composants.

3.3.2.8 COPEX-Chimie

Copex-chimie [74] est une application web dans laquelle les apprenants doivent déterminer la concentration du colorant rouge dans un sirop de grenadine par titrage spectrophotométrique. Pour atteindre cet objectif, les apprenants doivent spécifier une procédure expérimentale qui sera simulée par l'application. La procédure expérimentale est structurée en trois étapes (préparation d'une série de solutions étalons ; obtention des points de la courbe standard ; détermination de la concentration du colorant dans le sirop de grenadine). Pour chaque étape, l'apprenant sélectionne les actions adéquates parmi une liste de huit actions (rincer un équipement, faire une dilution, etc.). Pour chaque action ajoutée à l'étape de la procédure expérimentale, les paramètres décrivant l'action doivent être définis par l'apprenant. Un tuteur artificiel est accessible sur demande pour évaluer la procédure. Ce tuteur évalue la procédure, étape par étape, en suivant un système basé sur des contraintes, et signale les erreurs à l'apprenant. [47][37]

Copex-chimie produit en sortie deux fichiers texte en format PDF dont l'un sous forme de protocole (ensemble d'étapes) en chimie contenant la liste des éléments utili-

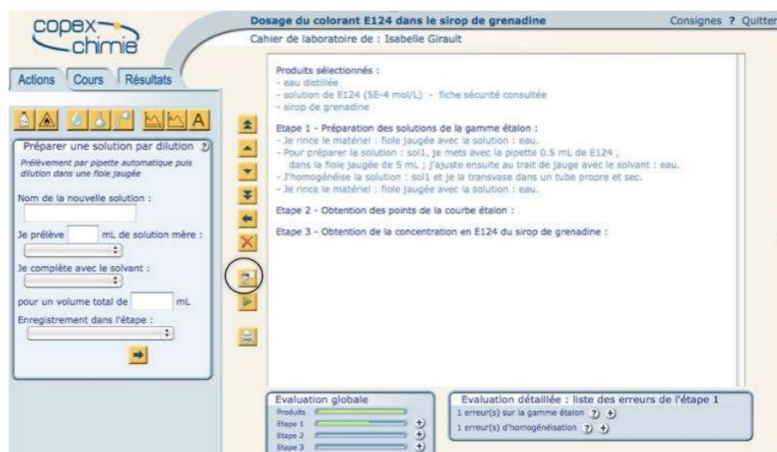


FIGURE 3.8 – Interface de COPEX-Chimie

sés ainsi que la liste des étapes à suivre pour arriver à l'objectif voulu. Le texte n'a pas une forme modèle quoi que les étapes sont organisées en lignes sans temporisation. Le deuxième fichier est divisé en deux parties, la première contient un ensemble d'informations (date, temps, action, paramètres et détail) organisées en colonnes et les lignes correspondent aux actions élémentaires (observés). La deuxième partie est un ensemble d'indicateurs calculés dans plusieurs sessions pour chaque apprenant. Les coordonnées de l'apprenant sont mentionnées en en-tête dans les deux fichiers.

3.3.2.9 AMBRE-add

AMBRE-add est un EIAH destiné à être utilisé régulièrement par des élèves de primaire pour apprendre le domaine des problèmes additifs en mathématiques. L'EIAH AMBRE-add met en œuvre le cycle du projet AMBRE pour le domaine des problèmes additifs. Chaque étape du cycle peut elle-même être devisée en plusieurs sous-étapes. A chaque moment, l'apprenant peut demander de l'aide ou un diagnostic sur sa réponse (bouée de sauvetage et feu tricolore, ...) demande qui donnera lieu à un message d'explication. Ce diagnostic est systématique à la fin de chaque étape du cycle. Les messages d'explication textuels ou graphiques utilisés indiquent bien le problème dans les erreurs commises. L'apprenant peut également via le menu accéder à un certain nombre de fonctionnalités, dont plusieurs outils de calcul destinés l'aider à effectuer le calcul demandé, ce calcul n'étant pas l'objectif principal de l'apprentissage. AMBRE-add est basé sur le RAPC (Raisonnement A Partir de Cas), il se compose essentiellement d'une interface graphique et d'une base de connaissances. [48] [78]

AMBRE-add produit des traces sous forme d'un seul fichier en format XML suivant un schéma décrit avec un fichier de métadonnées en format XSD. Le fichier XML contient dans l'ordre : des informations de caractéristiques des traces, des informations

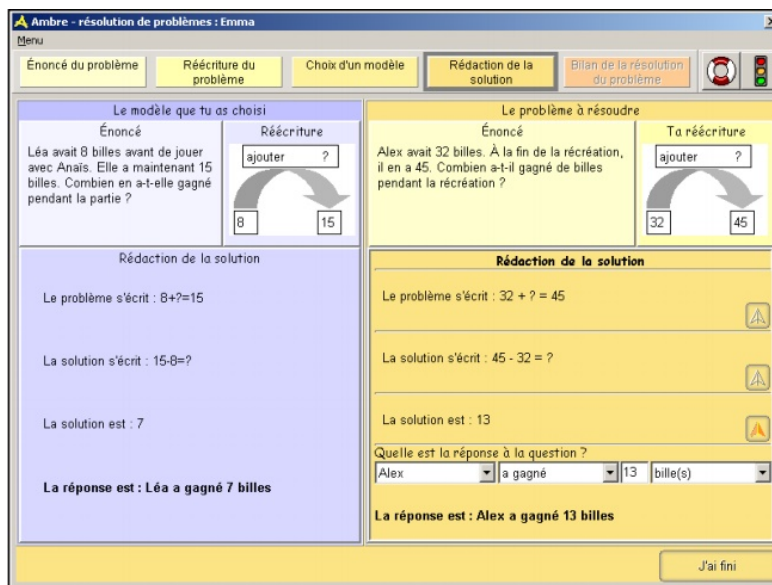


FIGURE 3.9 – L'étape d'adaptation dans AMBRE-add

sur l'activité à faire et la suite des sessions parcourues. Pour chaque session, le fichier enregistre les caractéristiques et les éléments de la trace.

3.3.2.10 ColAT

ColAT (Collaboration Analysis Tool) discuté plus en détail dans [8] est un outil d'analyse de la collaboration dans les environnements d'apprentissage. L'outil ColAT utilise la forme d'une scène de théâtre comme métaphore et cadre d'organisation. Selon cela, on peut observer l'action en suivant l'intrigue de différents points de vue. La vue des événements permet d'étudier les détails de l'action et de l'interaction, la vue des tâches permet d'étudier des segments d'action déterminés, tandis que la vue des objectifs étudie l'activité au niveau stratégique, où les processus cognitifs des acteurs sur la collaboration sont plus clairement décrits. La possibilité de visualiser un processus à l'aide de divers médias (vidéo, audio, texte, fichiers journaux, images fixes), à différents niveaux d'abstraction (événement, tâche, objectif), constitue une approche innovante. Il combine en un seul environnement l'analyse hiérarchique d'une activité collaborative au caractère séquentiel des données d'observation.

ColAT produit un fichier XML fondé sur un modèle théorique de traces. Le fichier comporte deux parties : une partie contexte et une partie actions.

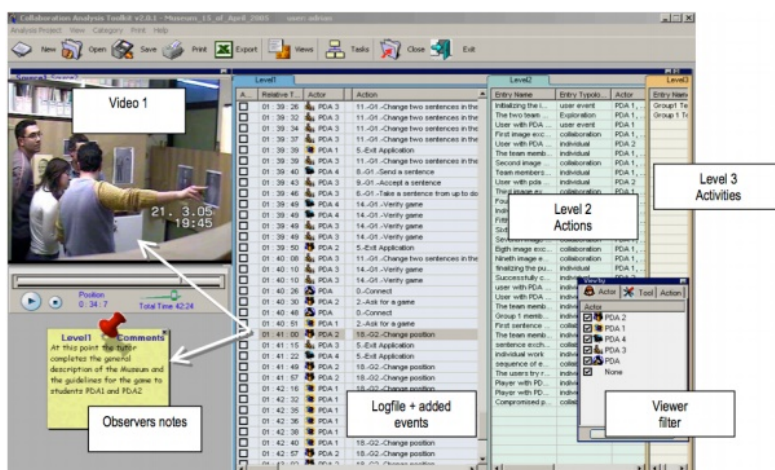


FIGURE 3.10 – Une fenêtre de ColAT

3.3.2.11 PEPITE

PEPITE [53] s'intéresse au diagnostic des compétences des élèves en algèbre à la fin du collège. Il se fonde sur une étude didactique et cognitive. Le logiciel comporte trois modules : PépiTest propose des exercices aux élèves et recueille leurs réponses, PépiDiag analyse 80% des réponses des élèves et les codes en référence au modèle de compétence, PépiProfil construit un profil de l'élève à partir d'une analyse transversale des résultats du codage des réponses sur l'ensemble des exercices. Ce dernier module permet à l'enseignant de visualiser le profil mais aussi de le compléter et de le modifier.

Les traces produites par PEPIT sont enregistrées dans trois fichiers textes. Le premier contient d'une manière successive et sans temporisation, les informations de la session et les productions de l'élève pour chaque exercice ou partie d'exercice sans les indiquer (l'identification des valeurs de production est implicite tout dépend du numéro et de la partie de l'exercice). Le deuxième fichier contient une liste temporisée des activités de haut niveau de l'élève du genre : faire exercice, Enregistrer, Effacer, Utiliser calculatrice, Utiliser brouillon, etc. Le troisième présente des résultats d'analyse des fichiers précédents du genre : Fréquence et temps pris pour les actions, séquence d'exercice.

3.3.2.12 TRI

TRI (Tri et Recyclage Interactifs) [1] est un logiciel de sensibilisation au tri sélectif et au recyclage destiné à des jeunes enfants ne sachant pas forcément lire : toutes les consignes et informations sont données oralement. Il est constitué de plusieurs activités, cours et exercices qui comportent plusieurs niveaux, des jeux ainsi qu'une aide. Il est possible de définir une séquence personnalisée pour chaque élève, mais cette per-

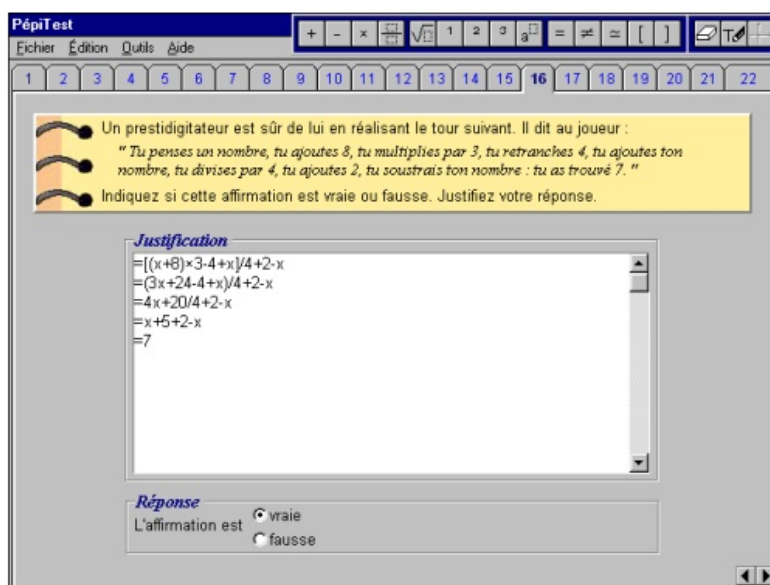


FIGURE 3.11 – Réponse à un exercice dans PEPITE

sonnalisation doit être faite manuellement. Le travail de l'enfant étant enregistré dans un fichier de profil qui fournit un bilan à l'enseignant. [55]

TRI produit un fichier texte sert comme profil contenant l'historique de plusieurs séances successives avec l'identifiant de l'élève en en-tête. Chaque ligne dans chaque séance correspond au couple : (Temps, Action) avec éventuellement des paramètres ou des détails pour quelques types d'actions. Le fichier trace ou profil n'a pas alors presque une structure fixe bien claire.

3.3.2.13 E-Lycée

E-Lycée [3] propose un environnement de travail collaboratif synchrone particulièrement riche (tableau blanc, chat, push web, visualisation synchrone de documents audio-visuels, awareness vidéo, discussion audio, traçage réflexif individuel et collectif) comme l'illustre la figure suivante. En se basant sur les traces générées par E-Lycée, chaque apprenant engagé dans un travail collectif dispose de fonctions pour comparer ses processus avec celui des autres, pour mieux comprendre comment un autre apprenant en est arrivé à une proposition collective suite à une activité individuelle ou avec d'autres. [85][35]

Les traces produites par E-Lycée sont transformées en traces plus agréables et visualisées pour être facilement utilisables d'une façon réflexive par les apprenants. E-Lycée présente les traces sous forme de trois éléments : Auteur, Action et Entité. L'auteur définit l'acteur intervenant dans l'action car E-Lycée est un outil collaboratif impli-



FIGURE 3.12 – Un jeu dans le logiciel TRI

quant plusieurs acteurs. Les actions peuvent être de type : Lire, Editer, Consulter, etc. Les entités peuvent être de type : Texte, Dictionnaire, Chat en ligne, etc. Ces éléments sont liés par des relations hiérarchique et sémantiques du genre : Auteur1-Editer-Texte, Auteur2-Consulter-Dictionnaire, etc.

3.3.2.14 Tables au trésor

Tables au trésor [2] est un logiciel d'entraînement aux tables mathématiques. Il est constitué de plusieurs types d'exercices entièrement paramétrables pour correspondre aux besoins des enseignants en s'adaptant à la progression dans le programme scolaire. Il est possible pour l'enseignant de définir une séquence personnalisée pour chaque élève à l'aide d'une interface dédiée. Cette interface permet de spécifier très précisément la séquence en contraignant les exercices qui la composent, mais aussi le type de rétroactions proposées. [54]

Le travail de l'élève étant enregistré dans un fichier de profil. Ce fichier qui sert donc de trace pour l'apprenant est un fichier texte contenant successivement et d'une manière semi structurée, la liste des sessions faites par l'élève et pour chacune, l'exercice fait, les opérations réalisées, le score, ... Les identifiants de l'élève ainsi que de son enseignant sont placés sur l'en-tête du fichier.

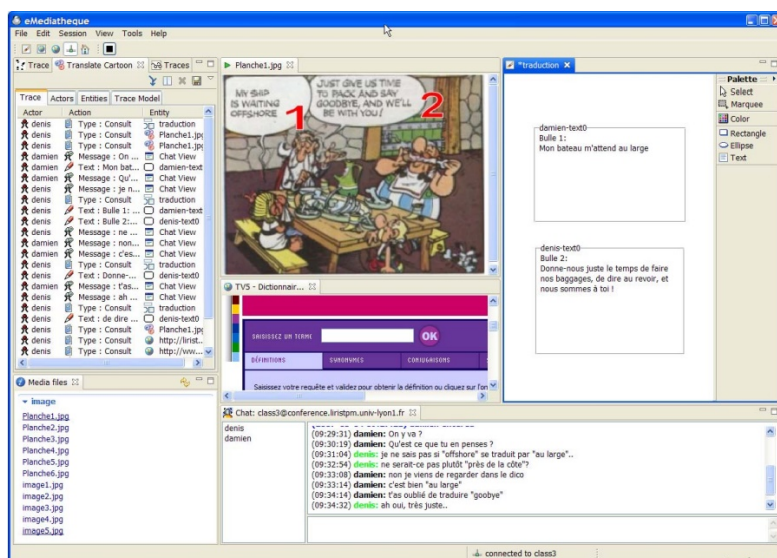


FIGURE 3.13 – Environnement d'apprentissage collaboratif E-Lycée

3.3.2.15 Moodle

Moodle [102] représente l'une des plates-formes e-learning open-source les plus utilisées qui permet de créer des sites Web de cours, en garantissant leur accès uniquement aux étudiants inscrits. Cette plate-forme permet l'échange d'informations entre utilisateurs géographiquement dispersés à travers des mécanismes de communication synchrone (chats) et asynchrone (forums de discussion). D'un point de vue fonctionnel, elle dispose de fonctionnalités facilement configurables, permettant la création de processus d'évaluation des étudiants (quiz, tests en ligne et enquêtes), ainsi que la gestion de leurs tâches avec leur calendrier et des outils complémentaires pour soutenir le processus d'enseignement et d'apprentissage. La plateforme Moodle se caractérise par un ensemble de fonctionnalités regroupées en deux classes différentes : ressources (site web, document Word, animation Flash, etc.) et modules (base de données, chat, forum, quiz, wikis, SCORM, etc.). En ce qui concerne les activités offertes par Moodle six classes sont à citer : création, organisation, prestation, communication, collaboration et évaluation. Grâce à son architecture modulaire, Moodle profite de plugins développés par sa communauté pour permettre l'extension de ses fonctionnalités et de répondre ainsi à des besoins spécifiques.

Les traces d'activité des apprenants utilisant Moodle se stockent dans une base de données relationnelle contenant environ 153 tables MySQL enregistrant des informations sur le timing, les actions des apprenants, les outils utilisées, etc. cette base de données est accessible au formateurs informaticiens directement ou via des scripts en PHP par exemple.



FIGURE 3.14 – Écran de choix d'activités de Tables au trésor

3.3.3 Résultats

La liste de logiciels pédagogiques que nous venons de présenter nous a permis d'avoir une idée sur l'existant des traces dans la littérature. Comprendre le contexte de génération des traces, l'objectif et les fonctionnalités essentielles des logiciels tracés nous paraît utile pour comprendre et analyser ces traces. Après avoir présenté d'une façon détaillée les caractéristiques de chacune des traces des systèmes étudiés nous voulons maintenant faire le récapitulatif en indiquant les points communs et les différences remarquées. Concrètement, le seul point commun autour duquel toutes les traces se mettent d'accord et qui les caractérise par définition est la considération de la temporisation parallèlement à l'enregistrement de l'interaction. Effectivement la temporisation était toujours présente explicitement sous plusieurs formats du type : dd :hh :mm :ss, 99 :99, 9999, etc. ou implicitement compris par la chronologie des événements. Beaucoup de divergences ont été remarquées dans les traces traitées et ainsi beaucoup de remarques ont été soulevées. Nous présentons dans ce qui suit les constatations qui nous a paru essentielles pour concevoir notre futur solution le plus pertinent et générique possible.

- Trois types de traces sont distingués comme étant les plus fréquemment utilisés les traces textes, XML et les bases de données. D'autres systèmes utilisent exclusivement ou pas des enregistrements multimédia pour tracer l'interaction. Pour chacun de ces trois types on a trouvé plusieurs variantes à savoir : plusieurs formats texte tel que ".txt", ".csv", ".pdf", des fichiers XML sans ou avec schéma explicite tel que ".dtd" et ".xsd", plusieurs bases de données relationnelles comme "MySQL" et "Postgre".
- La temporisation qui signifie l'information désignant le temps n'a pas un format



FIGURE 3.15 – La plateforme Moodle

- unique et pas toujours explicite. Comme déjà mentionné, plusieurs formats standards sont utilisés représentant ou pas le timing réel des actions enregistrées.
- Les séparateurs sont des caractères ou des chaînes de caractères délimitant des portions d'informations dans les traces texte. Dans les traces texte analysées plusieurs séparateurs ont été distingués, le plus utilisé est le ";" mais il y a aussi le caractère blanc, le ":", "| " et le caractère fin de ligne.
 - Les tailles des traces traitées étaient aussi très variées et ceci est expliqué par la différence des contextes, des nombres d'activités, des aspects de collaboration, etc. des logiciels traçant. A titre d'exemple, nous avons tombé sur des traces de quelques kilo d'octets et sur d'autres touchant des méga d'octets.
 - Les systèmes tracés n'adoptent forcément pas un type unique de traces, plusieurs fichiers éventuellement de formats différents peuvent en même temps être utilisés pour un seul logiciel. Plusieurs systèmes utilisent plusieurs fichiers texte pour enregistrer des catégories variées d'informations, d'autres utilisent des combinaisons du texte avec des enregistrements multimédia généralement pour des besoins d'analyse manuel et de renforcement.
 - Le format basique utilisé dans les log n'est pas toujours respecté par les systèmes traçant. Des injections et des réorganisations sont généralement faites. On parle ici des méta-informations concernant l'apprenant, l'activité, etc. On peut citer aussi les morceaux d'informations parachutées en vrac du genre : "intitulé texte1 texte2 ..." destinées à des utilisateurs humains. Des informations de configuration et de profil d'apprenant (généralement à l'entête du fichier) ou des récapitulatifs et bilans d'analyse (généralement au pied du fichier) peuvent être présentes dans la trace.
 - Les traces enregistrées ne concernent toujours pas un seul acteur, elles peuvent

- filmer l'interaction dans une activité collaborative et concerne plutôt un groupe d'apprenants.
- Les traces en format texte peuvent contenir des parties semi structurées et d'autres non structurées. Cet aspect va nous obliger à penser à une segmentation de ce genre de trace et appliquer des processus de traitement différents pour chaque partie.

3.4 Conclusion

Afin d'avoir une idée assez claire sur le format, la structure, le type et le contenu des traces d'interaction générées par les systèmes tracés nous avons voulu faire une étude d'analyse pour que nos choix soient les plus génériques et les plus standards possibles. En fouillant dans la littérature des EIAH et des systèmes tracés en générales nous avons pu choisir un nombre de logiciels devenant par la suite le sujet de cette étude. L'étude consistait à donner une brève explication sur le logiciel avec éventuellement des captures d'écran. Après consultation et analyse manuelle d'un ou plusieurs exemples de traces de chaque logiciel traité, une explication a été fournie. Cette explication s'intéressait aux formats de traces utilisés à savoir : texte, xml, base de données, enregistrement audio, vidéo, etc. Elle s'intéressait aussi au contenu des traces et ainsi les informations enregistrées tel que les informations de profil, les productions saisies par l'utilisateur, les informations de communication, etc. Dans le cas des traces non structurées des informations plus profondes sont fournies expliquant la structure de ces traces tel que les séparateurs utilisés et l'organisation interne.

4

Collecte et Importation des traces

Plan du chapitre

2.1	Introduction	10
2.2	Assistance à l'utilisateur en milieu d'apprentissage	11
2.2.1	Approches d'assistance	11
2.2.2	Moyens d'assistance	13
2.2.3	Besoins d'assistance	14
2.2.4	Dimensions d'assistance	15
2.2.5	Modalités d'intervention dans les systèmes d'assistance	17
2.2.6	Assistance contextuelle	18
2.3	Traces d'interaction	19
2.3.1	Définitions	19
2.3.2	Les traces dans la littérature	20
2.3.3	Modélisation des traces	21
2.3.4	Système à base de traces	22
2.3.5	La collecte dans les systèmes de bases de traces	24
2.4	Assistance à base de traces	24
2.5	Conclusion	26

4.1 Introduction

Les traces numériques sont des inscriptions temporalisées liées à un système informatique donné et ce pour diverses exploitations. Bien que la naissance des traces date de plusieurs dizaines d'années, elles ont subi une mutation importante cette dernière décennie. En effet, de nombreux systèmes informatiques ont commencé à produire et/ou exploiter les traces, mais chacun avec son propre format, son propre contenu et pour ses propres objectifs. Le concept de trace s'est bien développé, par rapport aux formats possibles adoptés, aux contenus considérés et aux exploitations visées. Cependant, des défis majeurs se posent : celui de leur compréhension et celui de leur interopérabilité. Le problème de compréhension se pose généralement durant les processus d'exploitation des traces ayant pour but la visualisation, l'analyse et la déduction de connaissances. Notre travail s'intègre dans le domaine de l'intégration de données, pour justement proposer une solution au problème de l'interopérabilité. Nous visons à intégrer des traces variées dans un formalisme unique pour permettre ensuite leur exploitation.

Un système tracé est un système capable de générer des traces numériques d'interaction liées aux activités qu'il propose à l'utilisateur. Les systèmes tracés actuels utilisent des traces numériques très variées, à des fins aussi variées. Après consultation et analyse des traces d'un ensemble de systèmes tracés, nous avons abouti à un ensemble de constatations présentées dans le chapitre précédent.

Devant ces constatations, la question est : comment permettre à un utilisateur d'un système tracé, qui possède des traces d'activités issues de son système, de les importer en tant que traces modélisées (traces associées à leurs modèles explicites) dans un système adoptant un unique formalisme de représentation des traces, afin de pouvoir profiter des fonctionnalités de visualisation et d'analyse d'un tel système ?

Le problème de la collecte des traces a été largement abordé dans les travaux sur les traces d'interaction. En effet, la collecte est une étape critique dans le processus logique de traitement des traces. Son cas le plus usuel consiste à instrumenter les systèmes tracés afin de produire des traces de format et structure spécifiques [71] – ou une combinaison de types de trace dans certains cas comme celui de [56]. À la différence de ces situations où la collecte et l'exploitation des traces sont toutes deux assurées par le même système, il existe d'autres situations où ces deux traitements sont séparés. Dans un travail comme celui de [26] ou [90], le rôle du système tracé se résume à la collecte. Les traces collectées sont destinées à des plateformes dédiées au stockage et à l'analyse [20] ; [30]. Notre travail s'intègre dans ce dernier processus et essaye de résoudre le problème de l'importation de traces variées dans l'une des plateformes de gestion des traces appelée kTBS (kernel for Trace Based Systems) [81]. Ainsi, cette plateforme est le système de gestion de traces dans lequel nous souhaitons importer des traces venant de systèmes divers. Par ailleurs, nous souhaitons offrir cette fonctionnalité à n'importe quel utilisateur du système tracé qui souhaiterait en exploiter les traces, y compris si

il/elle n'a pas de compétences en programmation. En ce sens, nous nous démarquons de travaux similaires [33] ayant une approche plus technique. Ainsi, nous présenterons dans ce chapitre notre processus d'importation de traces variées dans kTBS et l'outil xCollector qui le concrétise. Ce processus a l'avantage de traiter des formats variés de traces (texte, XML et BDDR). Son principe est basé sur l'acquisition de l'expérience de l'utilisateur en matière de collecte de données éparpillées dans les traces existantes [15]. xCollector n'est pas lié à un système tracé spécifique, nous pouvons le considérer comme étant un générateur de collecteurs pour kTBS.

Dans la section suivante, nous détaillerons notre processus d'importation de traces en expliquant chacune de ses étapes : réécriture, création du modèle de trace et Mapping. En section 3 nous présentons xCollector, l'outil matérialisant ce processus, nous décrivons ses fonctionnalités essentielles et nous présentons un cas réel de son utilisation. La section 4 sera réservée à l'expérimentation faite sur l'outil xCollector et ainsi l'évaluation du processus proposé. La section 5 sera la conclusion de ce chapitre.

4.2 Processus Collecte et importation de traces

A l'instar du LIRIS, l'exploitation des traces intéresse de plus en plus d'utilisateurs dans des disciplines variées. En effet, les traces sont largement considérées comme des sources de connaissances et de raisonnement. Ainsi, dans une perspective de capitaliser les traces, d'en faire des déductions et des interprétations, les SGBT sont proposés. Si les potentialités des SGBT sont suffisantes pour motiver leur utilisation, leur accessibilité et leur utilisabilité pourraient rebuter certains utilisateurs. Ce travail est donc une manière de faciliter la tâche aux utilisateurs (qui ne sont pas forcément informaticiens) pour importer leurs traces dans le SGBT kTBS.

Pour chaque système tracé, le processus d'importation de traces que nous proposons [13] est composé de deux étapes : une étape de conception et de création (génération) de collecteur et une étape d'utilisation de ce dernier (l'importation en elle même). Trois types de traces sont gérés par ce processus : texte, XML et BDDR.

4.2.1 Génération du collecteur

Le rôle essentiel des collecteurs que le générateur doit produire est de construire des traces importables à kTBS à partir des traces existantes. Les traces importables sont des traces conformes et associées à des modèles kTBS de traces. Le problème revient donc à créer un modèle de trace pour chaque collecteur et à construire des traces qui lui sont conformes à partir de celles fournies en entrée. La question qui se pose est : comment former des traces kTBS à partir d'exemples de traces externes ? Pour répondre à cette question, on propose de définir des règles de transformation qui vont permettre de chercher et calculer les éléments de la trace kTBS à partir de la trace fournie en entrée.

La définition de règles de Mapping entre les éléments de la trace externe et les éléments du modèle de trace est à la base de ce processus. Le Mapping sert ainsi à définir les correspondances entre tous les éléments du modèle kTBS de trace et leurs instances dans les traces en entrée. Il doit être effectué de la façon la plus générale possible pour qu'il reste pertinent quelles que soient les traces introduites. Afin de pouvoir créer des modèles kTBS de traces accompagnés des règles de Mapping capables de convertir des traces externes en traces importables respectant ce modèle, un processus en trois étapes est proposé : la réécriture, la création du modèle et le Mapping (Figure 1).

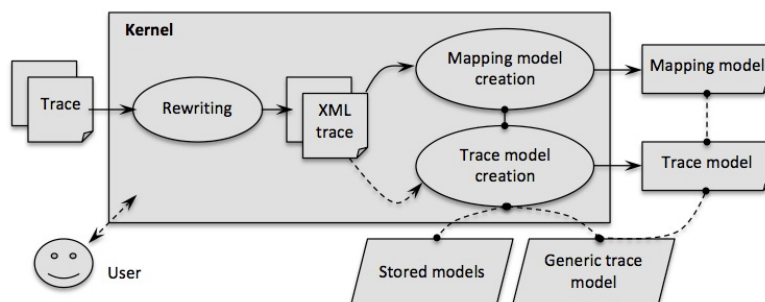


FIGURE 4.1 – Processus de génération des collecteurs

4.2.1.1 Réécriture

Compte tenu des constatations effectuées en introduction, trois formats de traces sont essentiellement distingués. Pour simplifier le processus d'importation des traces à kTBS et éviter un traitement spécialisé pour chaque format, nous avons choisi de considérer XML comme format pivot. Autrement dit, les traces dans d'autres formats seront réécrites en XML, ce qui va ainsi unifier le processus d'importation. XML [104] est un standard largement utilisé pour sérialiser des contenus structurés (processus de transformation des objets en flux de données). Ce qui motive aussi le choix de XML, ce sont les possibilités qu'il offre en termes d'outils d'interrogation, de transformation et de repérage. La structure arborescente d'XML rend (à notre avis) son contenu facilement accessible et même plus lisible dans certains cas de traces.

La réécriture est une étape préparatoire par rapport au Mapping. Son objectif peut être, selon le format d'entrée, la structuration ou la restructuration des traces avant leur chargement. Ainsi, la réécriture va consister à :

- Structurer la trace dans un format XML s'il s'agit d'un texte : les lignes du texte deviennent des balises <ligne> XML dont les fils sont des balises <cellule> contenant chacune un des items de la ligne texte en question.
- Restructurer la trace dans un format XML s'il s'agit d'une source de BDDR (table ou requête) : les lignes de la source de BDDR deviennent des balises <ligne> XML dont

les attributs et leurs valeurs sont les noms des colonnes avec les valeurs de la ligne de la source BDDR en question.

4.2.1.2 Création du modèle kTBS de trace

D'après la définition donnée dans le chapitre 02, un modèle de trace dans kTBS est une description de trace exprimée dans un langage fondé sur RDF, définissant des types d'obsels, des types d'attributs d'obsels et des types de relations pouvant exister entre les obsels. Pour faciliter la rédaction du code RDF décrivant ces éléments du modèle par l'utilisateur, nous proposons un moyen graphique pour le faire. Dans cette même perspective d'aider l'utilisateur durant le processus de création du modèle, un exemple de trace externe peut être visualisé. La réutilisation de modèles kTBS existants ainsi que leur modification pour en créer d'autres est un moyen proposé pour construire facilement des modèles kTBS de traces.

4.2.1.3 Mapping

Dès que la trace chargée par l'utilisateur ou sa copie réécrite est visualisée, et dès que l'utilisateur commence à construire son modèle de trace (éventuellement guidé par le contenu de la trace) en définissant ses éléments, il peut parallèlement commencer à dresser des liens sémantiques entre les éléments de sa trace et des éléments de son modèle en construction. La sémantique de ces liens est de la forme "est de type". Si on suppose qu'un lien <eltXML - eltModel> est dressé, ceci s'interprète par "eltXML est de type eltModel".

Pour mieux justifier l'utilité de tels liens, il est utile de rappeler le but recherché, i.e. un ensemble de règles de conversion des traces externes en traces importables au kTBS. Ainsi, ces règles doivent être capables de repérer des instances des éléments du modèle dans la trace externe et aussi de les convertir en instances dans la trace résultat. Les liens que l'utilisateur est invité à faire ont donc pour objectif la définition des éléments pertinents dans la trace externe. Il reste au générateur de collecteurs à généraliser ces définitions pour prendre en compte les variations éventuelles entre les exemples de traces externes.

À ce propos, plusieurs travaux ont été menés dans le contexte de l'annotation dans le web, dont l'objectif était de générer des repères (expressions XPath) dans des pages web correspondant aux clics de l'utilisateur [49][59][61][84], qui soient robustes aux changements les plus fréquents dans la structure de ces pages web. Effectivement, XPath [105] - le premier outil XML de pointage - n'a pas seulement prouvé ses capacités de repérage mais aussi il en donne des choix multiples.

Dans notre contexte le problème devient : comment générer des expressions XPath correspondant au choix de l'utilisateur (éléments XML sélectionnés) qui restent valides pour les futurs exemples de traces XML ? Une expression <Exp> créée à partir d'un

élément sélectionné <Elt> qui correspond au type <Type> dans le modèle de trace est valide si et seulement si elle repère tous les éléments de type <Type>. Les liens que l'utilisateur doit dresser et que nous avons représentés par des couples <eltXML - elt-Model> auront la forme <ExpressionXPath - Type> où Type est un type d'obsel ou d'attribut dans le modèle.

Dans les travaux de Abe et Hori [50][4][5][49][51], plusieurs modèles d'expressions XPath sont discutés : - Les expressions "absolues" qui effectuent le repérage en suivant la hiérarchie du document XML de la racine jusqu'au nœud cible, avec la forme : /Racine/fils/ ... /eltSélectionné. - Les expressions "relatives" qui repèrent un nœud par rapport à des positions d'ancres stables, avec la forme : ExpressionAncre//eltSélectionné [Distance]. - Les expressions "conditionnelles" repérant les nœuds par le biais d'une condition à satisfaire, avec la forme : //eltSélectionné [Condition].

Quant à nous, nous introduisons la notion de "contexte" qui est à la base de nos modèles d'expressions XPath [13]. Un contexte est une expression XPath repérant un endroit voisin d'un élément instance (obsel ou attribut) du modèle de trace ou d'un autre contexte. Pour donner plus de robustesse à nos expressions, nous avons aussi proposé la combinaison d'expressions relatives et conditionnelles. Globalement, nos expressions ont la forme suivante : ExpressionContexte //eltSélectionné [Condition]. Nous avons utilisé ce modèle d'expression pour exprimer des repères, d'instances de types d'obsel, d'instances de types d'attributs et aussi pour définir des contextes.

Concrètement, après que l'utilisateur ait défini son Mapping entre un élément XML et un élément du modèle, notre système propose un nombre de suggestions XPath avec des conditions variées selon le contenu de la trace, pour désigner l'élément XML sélectionné. Ces conditions peuvent porter sur :

- La valeur de l'élément sélectionné, par exemple : //N [text() = "Exercice"] pour choisir tous les nœuds N dont le texte est égal à "Exercice".
- Les valeurs des attributs de l'élément sélectionné, par exemple : //N [@action = "Envoyer Msg"] pour choisir tous les nœuds N dont l'attribut "action" égale "Envoyer Msg".
- La valeur du texte ou des attributs des fils de l'élément sélectionné, par exemple : //N [./time/@begin = "10 :00"] pour choisir tous les nœuds N dont l'attribut "begin" du fils "time" est égal à "10 :00".
- Le type de valeur de l'élément sélectionné, par exemple : //N [matches(@option, Numérique)] pour choisir tous les nœuds N dont le type de l'attribut "option" est numérique.
- La position de l'élément sélectionné, par exemple : //P/N [2] pour choisir tous les nœuds N dont la position est égale à 2 par rapport au nœud parent P.

Les propositions XPath repérant un élément sélectionné par l'utilisateur sont générées par l'algorithme présenté ci-dessous. Son principe est de générer des expressions XPath prenant en compte le nom de l'élément sélectionné, sa valeur, le type de sa valeur,

sa position, les valeurs de ses attributs et le contenu de ses fils.

Algorithm 1 Génération des expressions XPath

```

ExpName  $\leftarrow \phi$ 
Conditions  $\leftarrow \phi$ 
ExpName.Create("/" + SelectedElt.name)
if (SelectedElt.attributes  $\neq \phi$ ) then
  for (each attribute in SelectedElt.Attributes) do
    Conditions.addValue(attribute.value)
  end for
else
  Conditions.addValue(SelectedElt.value)
end if
for (each child in SelectedElt.Children) do
  if (child.Attributes  $\neq \phi$ ) then
    for (each attribute in child.Attributes) do
      Conditions.addValue(attribute.value)
    end for
  else
    Conditions.addValue(child.value)
  end if
end for
Conditions.addType(SelectedElt.valueType)
Conditions.addPosition(SelectedElt.position)
Expressions.add(ExpName)
for (each condition in Conditions) do
  Expressions.add(ExpName + condition)
end for

```

Cet algorithme produit des expressions de la forme : //eltSélectionné [Condition]. L'utilisateur aura le choix d'utiliser un contexte pour repérer sa sélection ou pas. Si un contexte est utilisé, le repérage de l'élément sélectionné devient relatif à ce contexte et l'expression devient : ExpressionContexte//eltSélectionné [Condition]. À son tour, ExpressionContexte a la forme : [ExpressionContexte'] //eltSélectionné', etc. Dans certains cas de traces, la combinaison de plusieurs types de conditions peut avoir de la valeur afin de concevoir des expressions pertinentes. Selon le contenu des traces traitées, on peut imaginer des expressions XPath : (1) avec une seule condition (avec/sans contexte) ou (2) avec plusieurs conditions combinées (avec/sans contexte).

En effet, les expressions XPath sont proposées à l'utilisateur après une transcription en langage naturel. Pour choisir une proposition ou un contexte donné, on propose aussi à l'utilisateur de se fonder sur l'effet de son choix d'expression. Autrement dit, quand l'utilisateur choisit une (ou plusieurs) propositions, le système lui indique les

éléments repérés par cette (ou ces) proposition(s), ce qui permet à l'utilisateur de juger de la pertinence de son choix. Ce choix est donc fondé d'une part sur la formulation en langue naturelle des expressions XPath et d'autre part sur l'effet de ces expressions pour repérer les éléments de la trace que l'utilisateur souhaite récupérer.

À travers un ou plusieurs exemples, l'utilisateur va pouvoir créer son modèle kTBS de trace, définir tous les Mapping nécessaires et donc choisir toutes les expressions efficaces qui vont servir à la conversion des traces externes en traces importables. Les couples <ExpressionXPath - Type> (le modèle de Mapping) ainsi produits et le modèle kTBS de trace forment une partie importante du collecteur cherché.

4.2.2 Utilisation du collecteur

Les modèles de Mapping et de traces sont des unités statiques incapables à elles seules de convertir ou de collecter des traces. Le Module Kernel intégré dans notre modèle sert à compléter ces deux modèles pour former de véritables collecteurs. Le rôle du Kernel est d'assurer deux fonctionnalités : (1) la réécriture des traces externes en traces XML et (2) l'utilisation du modèle de Mapping pour créer des traces kTBS puis les associer à leur modèle pour qu'elles soient importables. Les collecteurs résultants sont ainsi définis par le triplet : <ModelMapping - ModelTrace - Kernel> (Figure 2).

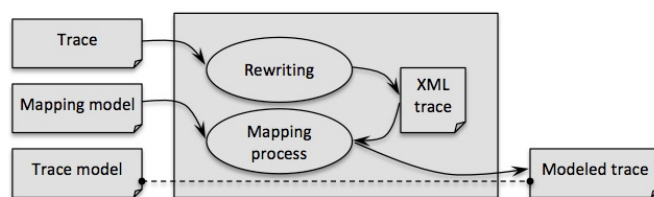


FIGURE 4.2 – Utilisation du collecteur

À la présentation d'une nouvelle trace externe, le kernel identifie le type de la trace puis procède éventuellement à sa réécriture en exploitant les informations de réécriture (les séparateurs, le type de codage du texte, le chemin de la BDDR, etc.) déjà définis lors de la création du collecteur. Le kernel commence ensuite à utiliser le modèle de Mapping et à chercher des occurrences de chaque type appartenant au modèle de Mapping. Il parcourt les types du modèle et en cherche toutes les occurrences présentes dans la trace. La recherche est simplement faite en lançant les expressions XPath correspondantes. Chaque occurrence trouvée est affectée à la valeur de l'obsel ou de l'attribut correspondant au type en question. La trace kTBS se construit ainsi progressivement en détectant tous les obsels et attributs existants dans la trace et en en déduisant les valeurs correspondantes. De cette façon le collecteur construit des traces kTBS conformes à leurs modèles et prêtes à être importées au kTBS.

4.3 L'outil xCollector

4.3.1 Présentation

xCollector est un outil java destiné aux utilisateurs voulant importer à kTBS des traces texte, XML ou BDDR. Via son interface graphique, xCollector permet de créer des modèles kTBS de trace et de générer des repères (expressions) XPath pour identifier, dans les traces externes, les éléments nécessaires pour créer de nouvelles traces importables conformes à leurs modèles. xCollector propose diverses fonctionnalités [14] :

- Le chargement des traces externes : les traces de l'utilisateur sont chargées et visualisées sous forme d'arbre XML avec un habillage proche de l'original. xCollector affiche les traces texte sous forme de lignes de texte et les traces BDDR sous forme de lignes de champs identiques. Lors du chargement, des données doivent éventuellement être renseignées : les séparateurs pour les traces texte, la requête/la table et la BDDR pour les traces stockées dans des BDDR.

- La création du modèle kTBS de trace : sans se soucier de coder en RDF ni de connaître la syntaxe kTBS pour définir des modèles de trace, l'utilisateur peut facilement créer son modèle de trace en créant des types d'obsels, d'attributs et de relations. xCollector affiche sous forme arborescente le modèle de trace et permet la sélection de ses éléments. A tout moment, il est possible de générer et d'accéder au modèle créé au format RDF-turtle.

- La création du modèle de Mapping : ce processus doit être initié par l'utilisateur en établissant des liens entre des éléments de la trace et des éléments du modèle. Un lien est défini si l'utilisateur sélectionne, d'un côté un élément XML de la trace et de l'autre côté un élément de l'arbre du modèle.

Selon le type, le contenu et le voisinage de l'élément de trace sélectionné, le système propose un ensemble de références pour cet élément. Afin de connaître les éléments XML repérés par une référence donnée et pour faciliter ainsi le choix de celle-ci, xCollector utilise une métaphore de drapeau levé sur les éléments correspondant à cette référence. L'utilisateur choisit alors la proposition si et seulement si un drapeau est levé sur toutes les instances du type (d'obsel ou d'attribut) en question. En effet, chaque proposition est une expression XPath avec une condition différente. Le nombre et le type de ces conditions dépend alors du type, du contenu et du voisinage de l'élément sélectionné. Si, par exemple, le nœud sélectionné contient un attribut, une expression formée d'une condition sur la valeur de cet attribut est générée. Si cette valeur est numérique, une autre expression considérant ce type de valeur est générée. S'il y a plusieurs nœuds frères du nœud sélectionné ayant le même nom, une expression prenant en compte l'ordre de l'élément sélectionné est générée, etc.

L'utilisateur a aussi le choix de repérer un élément sélectionné par rapport à un contexte précédemment défini. Une expression correspondant à un type d'obsel ou

d'attribut donné peut être utilisée en tant que contexte. En dehors des types du modèle, l'utilisateur peut définir des contextes propres pour améliorer le repérage. Leur création suit les mêmes étapes : sélection, utilisation d'un contexte, choix de la proposition. Après sa génération, l'expression XPath, couplée avec le type sélectionné du modèle, est stockée dans un fichier XML représentant le modèle de Mapping.

4.3.2 Exemple d'utilisation

Le système GeoNotes [90] produit des traces textuelles composées de lignes contenant chacune une liste d'informations séparées par des virgules. Ces informations correspondent aux traitements effectués sur des images géographiques et ce à des fins pédagogiques. Le nombre et le type des informations est variable dans chaque ligne.

Après le chargement d'une trace GeoNotes, xCollector l'affiche dans son interface sous forme d'arbre XML mais avec une apparence proche du format texte (cf. Figure 3). Pour pouvoir faire des Mapping, au moins un élément doit être créé dans le modèle de trace. Supposons la création d'un type d'obsel appelé "zoom" : pour le faire il suffit d'introduire l'identificateur "zoom" ; et éventuellement un super type (s'il y a héritage). Ce type d'obsel nécessite un attribut "chemin" pour stocker le chemin de l'image à zoomer. En sélectionnant l'obsel "zoom", l'insertion d'un attribut "chemin" se fait en introduisant son identifiant, label et type (Figure 3).

Il y a alors deux éléments (zoom et chemin) pour lesquels on peut définir des Mapping. Afin de repérer le type "zoom", il suffit qu'on choisisse un des éléments XML de la trace correspondant à ce type, et le système génère les propositions d'expressions XPath. On choisit dans ce cas la proposition qui prend tous les éléments dont l'attribut txt contient l'information "zoom" (drapeau vert dans la Figure 4.3). L'utilisateur confirme l'enregistrement de la proposition (expression XPath) choisie en tant que règle de Mapping pour le type "zoom" (apparaissant dans la Figure 4.3).

Pour utiliser le type "zoom" comme contexte pour le type d'attribut "chemin", il doit être chargé afin de former une expression de contexte pour le type "chemin".

Après sélection d'un élément XML exemple pour le type "chemin", des propositions sont encore générées, parmi lesquelles l'utilisateur choisit celle qui prend tous les nœuds fils du contexte "zoom" nommés Cell et se trouvant en quatrième position (drapeaux rouges dans la Figure 4.3). Ainsi, les drapeaux jaunes indiquent les contextes (obsels "zoom"), les drapeaux rouges indiquent les attributs "chemins". Les règles de Mapping et le modèle kTBS de trace sont visualisés dans la partie droite de l'interface.

4.4 Expérimentation

Afin de tester son efficacité et son utilisabilité, xCollector a été testé par un nombre d'utilisateurs souhaitant importer des traces variées dans kTBS. Ces tests ont été

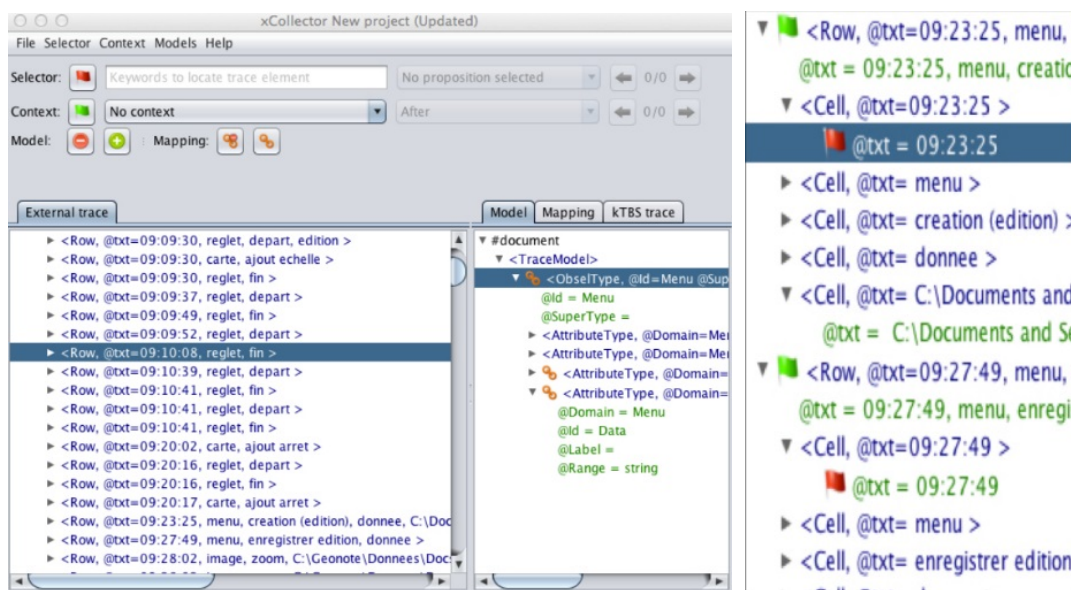


FIGURE 4.3 – xCollector

conduits auprès de six utilisateurs différents, à des moments et des lieux différents. Ces six utilisateurs n'étaient pas tous des informaticiens et ils n'étaient pas prêts et/ou capables à programmer des collecteurs pour importer leurs traces dans kTBS. Les trois formats de traces ont été testés : texte, BDD et XML. Les traces textuelles sont celles des EIAH « SuperViseur [67] », « ASKER [24] », « Aplusix [27] », « MTAT [77] » et « Classcraft [89] ». Parmi ces traces textuelles, certaines sont bien structurées, comme les fichiers « .CSV » et d'autres sont moins ou pas structurées, comme les fichiers « .TXT ». Une seule BDDR a été interrogée pour extraire des données de traces, celles de « ASKER [NG3] ». Pour importer des traces, des requêtes SQL ont été utilisées après avoir configuré l'accès à la base de données. Les traces en format XML issues de deux logiciels différents ont été aussi importées, celles du logiciel « Ambre-add [78] » et celles du « GeoNote [90] », les traces de AMBRE-add ayant une structure beaucoup plus compliquée que celles de GeoNote. Concernant les tailles des traces testées, elles varient entre des petites (Ambre-add, MTAT), moyennes (Aplusix, Classcraft, GeoNote) et grandes tailles (Superviseur, ASKER).

Durant les sessions de test, les utilisateurs ont procédé :

au chargement de leurs traces, après avoir éventuellement configuré les séparateurs à considérer (pour les traces textuelles) ou l'accès aux données (pour les traces en BDDR), à la création des modèles de traces, au repérage de données dans la trace ré-écrite et affichée sous forme XML, en utilisant éventuellement la notion de contexte, puis au Mapping des éléments repérés avec les éléments des modèles de traces.

Les utilisateurs ont pu pendant ces tests utiliser les fonctionnalités principales de

xCollector, et en général avec succès. En effet, xCollector a été progressivement amélioré suite aux commentaires et suggestions des utilisateurs, et notamment en ce qui concerne la facilité d'utilisation. Cependant, xCollector avait un problème de performance. Nous avons en effet remarqué une lenteur (parfois considérable) au chargement des traces de grande et moyenne tailles et au moment du repérage des éléments sur la trace ré-écrite. Nous avons pallié à ce problème en donnant la possibilité aux utilisateurs de limiter la taille de leurs traces, et ce en phase de configuration de leur chargement. [NG5] Bien que cette solution règle le problème en phase de création des collecteurs, la lenteur pourrait handicaper l'utilisation des collecteurs, notamment pour des volumes importants de traces, ce que nous devons prendre en compte pour la suite de notre travail.

Nous avons rencontré un problème particulier avec l'EIAH Aplusix : celui de la répétition de certains types dans le modèle de trace, ainsi que de la répétition de leurs Mappings correspondants. C'était le cas des traces d'Aplusix, où l'utilisateur était obligé de redéfinir les mêmes type d'attributs pour de nombreux types d'obsels. De plus, l'utilisateur était obligé de définir des Mappings identiques pour ces types d'attributs. Afin d'éviter cette répétition, nous avons proposé de donner la possibilité de définir des super-types cumulant les types redondants. Nous travaillons actuellement sur la mise en œuvre de ce super typage pour les types et aussi pour leurs Mappings.

Parmi les problèmes qui nous ont obligé de réfléchir davantage sur les possibilités de repérage offertes est celui des types composés d'attributs (cas d'ASKER). De tels types nous obligent à concaténer plusieurs données en utilisant plusieurs Mappings, fonction qui est actuellement offerte par xCollector. Vu la complexité de repérage dans les traces de « ASKER » en plus de l'utilisation de compositions variables de données pour certains de ses types d'attributs, nous travaillons à améliorer et renforcer les possibilités de repérage de notre outil. Pour le même logiciel nous citons un cas qui sort effectivement de notre cadre de recherche qui est la possibilité d'avoir un nombre variable de types dans le modèle de trace. En effet ce besoin est même en dehors de la définition que nous adoptons pour les modèles de traces.

En résumé, xCollector a montré la validité de l'approche que nous avons choisie pour l'importation de traces, et a répondu aux principaux besoins des utilisateurs. Cependant d'autres améliorations et fonctionnalités restent à développer pour augmenter le nombre d'EIAH pour lesquels il pourra être utilisé.

4.5 Conclusion

Les SGBT sont des plateformes susceptibles d'intéresser beaucoup d'acteurs intervenant sur des systèmes tracés, étant données les fonctionnalités qu'ils proposent. Pour pouvoir profiter de ces fonctionnalités de traitement et d'analyse des traces, les utilisateurs doivent actuellement faire l'effort de programmer des collecteurs pour importer

TABLE 4.1 – Résultats du test des fonctionnalités essentielles de xCollector

Software	Superviseur	ASKER	Aplusix	MTAT	Classcraft	AMBRE-add	Geonote
Format	txt file	txt file+DB tables	csv file	txt file	csv file	xml file	xml file
Header row	Yes	No	No	Yes	Yes	No	No
Size	[500,5000]kb	[500,5000]kb	[50,200]kb	[20,100]kb	[100,500]kb	[50,200]kb	[100,200] kb
Import	***	***	***	***	***	***	***
Model creation	***	**	***	***	***	***	***
Locating	***	**	*	***	***	***	***
Response time	*	***	**	***	***	***	***
Usability	**	**	**	***	***	***	***

leurs traces dans un SGBT. xCollector, le fruit de notre travail, propose une manière interactive de construire des collecteurs propres aux systèmes tracés, sans recours à la programmation. xCollector a été conçu et réalisé suite à une étude d'un corpus de traces de systèmes variés. Nous avons ensuite exploité ce corpus pour effectuer des tests montrant la validité et la généralité de l'approche choisie. Nous avons évalué la généralité de xCollector en l'utilisant pour collecter les traces de systèmes qui ne figuraient pas dans notre corpus initial, et nous avons également évalué son utilisabilité auprès d'un ensemble d'utilisateurs souhaitant importer leurs traces dans kTBS. Ces tests ont été conduits auprès de six utilisateurs ne pouvant et/ou ne voulant pas programmer des collecteurs pour leurs logiciels. Nous avons testé l'importation des traces de quatre logiciels pour lesquels les traces étaient au format texte, deux au format XML et un au format BDDR. Les utilisateurs ont pu essayer les fonctionnalités essentielles de xCollector, comme le repérage des données dans les traces chargées et leur Mapping avec les éléments du modèle de traces. xCollector a montré la validité de l'approche que nous avons choisie pour l'importation de traces, et a répondu aux principaux besoins des utilisateurs. Cependant d'autres améliorations notamment en ergonomie sont à faire, pour faciliter encore la prise en main de xCollector par les utilisateurs visés. D'autres fonctionnalités restent aussi à développer pour augmenter le nombre de logiciels pour lesquels il pourra être utilisé.

5

Interprétation des traces d'interaction

Plan du chapitre

3.1	Introduction	28
3.2	Quelques repères pour les EIAHs	28
3.3	Etude et analyse des traces existantes	30
3.3.1	Objectif	30
3.3.2	Traces étudiées	31
3.3.3	Résultats	45
3.4	Conclusion	47

5.1 Introduction

Les Environnements Informatiques pour l'apprentissage Humain proposent des possibilités d'apprentissage très intéressantes à travers leur variété, accessibilité et interactivité. Dès son apparition ces systèmes n'ont pas cessés d'évoluer et d'intéresser d'avantage d'acteurs. Cependant, à la différence du cas résidentiel où l'enseignant peut détecter si un apprenant est ou pas sur la bonne voie et ce grâce à ses facultés d'observation, l'assistance automatique reste le maillon le plus faible dans les EIAH. En revanche ces outils nous fournissent une grande quantité d'informations comparée aux simples observations visuelles d'un enseignant. Les informations et les traces produites par la plupart des EIAH sont malheureusement volumineuses et de natures techniques pour être exploitées manuellement par les enseignants-analystes.

Dans notre position, il n'est pas question de comparer les EIAH aux systèmes résidentiels classiques mais plutôt leur confrontation afin d'en extraire les avantages. Il sera par exemple idéal d'offrir à cet enseignant-analyste un maximum d'indices claires renforçant ces observations pour des déductions plus fines. Ces indices claires et importants en nombre sont pratiquement faciles à extraire depuis les traces d'interaction générées par les EIAH. Beaucoup de travaux ont été réalisés dans ce contexte. il convient de mentionner ici les travaux de Tarek Djoued [38, 39] sur l'ingénierie d'indicateurs et son exploitation pour améliorer les processus d'apprentissage. Nous pouvons citer aussi [41] qui présente un modèle pour l'analyse des traces, [88] présentant l'analyse des traces numériques comme mode de représentation et outil d'intervention, d'organisation et de gouvernance et [21] qui présente une démarche d'analyse et d'interprétation de la trace des interactions de l'apprenant dans un dispositif d'apprentissage. Dans la même perspective nous citons aussi les travaux s'intéressant à la visualisation et l'analyse des traces via les artefacts graphiques tel que dans [35] [9] et [72].

Dans le cadre de notre travail, nous proposons de travailler à partir de traces d'un niveau d'abstraction assez bas. Nous souhaitons mettre en évidence les patterns issus des traces qui permettent à l'analyste-enseignant d'établir des déductions sur l'activité des apprenants. Nous voulons par ça passer des traces à caractère technique peu ou pas compréhensibles à des traces plus évoluées formées de patterns d'actions. Dans un second temps nous voulons passer de la même façon de ces patterns vers d'autres encore plus évoluées. La question que nous posons alors est comment faire ces passages et transformations des traces techniques brutes de bas niveau sémantique pour arriver à des traces d'un niveau sémantique facilement traitable par un enseignant-analyste? L'idée est de détecter des patterns effectifs à partir des traces obtenues. Afin de découvrir les patterns, nous préparons manuellement une base de patterns initiale et nous proposons une méthode d'appariement entre la trace brute et la base conçue. Nous nous baserons alors essentiellement sur la recherche de motifs et la détection des séquences qui reviennent fréquemment. D'autres questions inhérents ainsi soulèvent à savoir la description et la modélisation des traces et patterns, le choix de la mesure

de similarité, le cas de chevauchement de patterns et l'alimentation et l'évolution de la base de patterns. Toutes ces discussions vont être entre autres le sujet de ce chapitre.

Si la recherche de patterns est peu utilisée dans le domaine des EIAH, nous pouvons cependant citer les travaux de [16], [73] et [32]. Les premiers proposent un outil de recherche de patterns dans un ensemble organisé de données issues d'interactions en ligne. Les patterns résultats sont ensuite présentés au chercheur, ils constituent une aide, une phase préalable à l'analyse, en lui permettant de visualiser les séquences, schèmes récurrents. [73] travaillent sur les résultats d'exercices. Ils repèrent au moyen des patterns les étudiants « à risque » (risque de ne pas réussir un exercice si d'autres n'ont pas été réussis) et des comportements d'apprenants. Les travaux de Cheype concernent des traces hétérogènes issues d'activités collectives. L'objectif est de détecter des indices sur les suites d'actions menant à l'échec ou la réussite d'exercices. Dans les deux cas la recherche de patterns est utilisée pour aider l'enseignant, pour avoir une meilleure appréciation du travail des étudiants.

Dans la section suivante nous présenterons le simulateur CUSIM et les modèles de traces adoptés. CUSIM et son utilisation seront notre contexte de travail et il va nous servir donc de système traçant et d'une source génératrice de traces. Dans la section trois nous détaillerons notre approche basée sur l'appariement et la détection de motif fréquents pour l'interprétation des traces en général et celles de CUSIM plus précisément. Cette approche a été concrétisée par un outil d'analyse de traces appelé "LogAnalyser". La présentation de cet outil sera le sujet de la quatrième section et nous terminerons ce chapitre par une conclusion.

5.2 Le simulateur CUSIM

CUSIM (Central processing Unit SIMulation) est un logiciel de simulation d'une unité de calcul de type Intel développé dans le cadre d'un projet de licence à l'université d'Annaba. Ce simulateur intègre la possibilité d'enregistrer des traces d'utilisation en format texte. L'objectif principal de ce simulateur est de permettre aux utilisateurs (étudiants en cycle de licence) de connaître le fonctionnement de l'unité centrale de traitement d'un ordinateur. CUSIM propose quatre activités aux apprenants :

- Catalogue : contient des tutoriels sur l'utilisation du CUSIM et des cours sur le fonctionnement de la CPU.
- Démonstration : permet à l'utilisateur d'observer les différents composants de l'unité centrale de traitement et leur rôle (unité de commande, séquenceur, unité arithmétique et logique, compteur ordinal, registre d'instruction), par le chargement des exemples expliquant le fonctionnement de la CPU.
- Manipulation : permet à l'utilisateur de manipuler chaque composant de la CPU (UC, MC, UAL). Nous pouvons effectuer des opérations arithmétiques/logiques et exécuter les instructions du programme stocké en mémoire centrale et observer comment L'unité de commande dirige le fonctionnement de toutes les autres

- unités : UAL, mémoire, entrées / sorties, etc. en leur fournissant les signaux de cadence et de commande.
- Exécution : dans cette activité l'utilisateur devrait être capable d'écrire un programme et de l'exécuter.
 - Fichier Log : Les traces sont enregistrées dans des fichiers texte (fichier log au format .txt). Toutes les opérations de l'utilisateur sur l'outil CUSIM sont stockées ligne par ligne en respectant un format défini par le système. Chaque ligne représente un évènement ou action observable appelée "observé". Pour chaque observé CUSIM enregistre le temps, l'activité, sous activité et l'action. Suite à une session d'activité où l'apprenant a pu mobiliser plusieurs connaissances en manipulant plusieurs fonctions de CUSIM, une trace contenant plusieurs informations sur l'activité est générée. Comme montré dans la Figure 4.2 la trace enregistre d'une manière générale une suite d'évènements et pour chaque évènement son temps de production, le parcours suivi ou les étapes de navigation dans l'outil pour arriver à la t'activité faite (Activité/ SousActivité/ SousSousActivité/ ...) et aussi la description de la tâche réalisée.

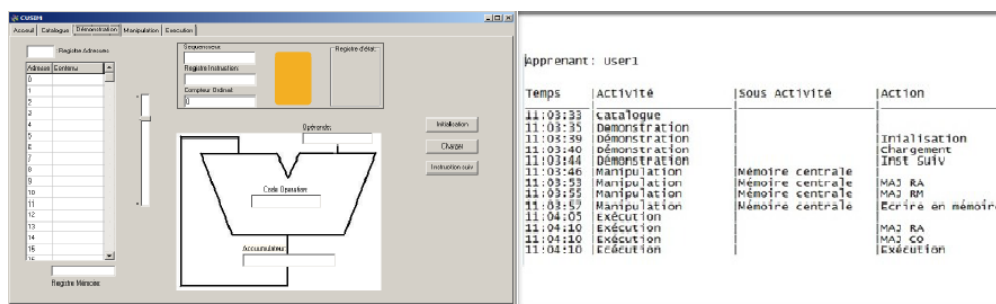


FIGURE 5.1 – Une interface de CUSIM et un exemple de log généré

5.2.1 Modèle de trace pour simulateur

Dans [12] nous avons proposé un modèle générique de trace produit par des systèmes de simulation. Ce modèle est spécifique en ça partie pour qu'il prenne en compte les spécificités des système de simulation. En analysant les possibilités d'interaction et de comportements de la part de l'apprenant et du simulateur lui même, il est possible d'extraire une collection d'objets formant le vocabulaire des éléments observables (évènements ou actions) possibles dans un simulateur. Cette collection peut être arrangée dans des classes, qui font aussi partie d'autres classes, etc. le tout forme une structure hiérarchique pouvant être implémenté par une ontologie. Cette hiérarchie représente exactement le modèle de trace dont on a déjà parlé (Figure 5.3).

- Indicateur : une quantité statistique signifie une durée, une fréquence ou autre, calculée à partir d'autres données brutes.
- Trace brute : suite temporelle d'éléments observables (observés).

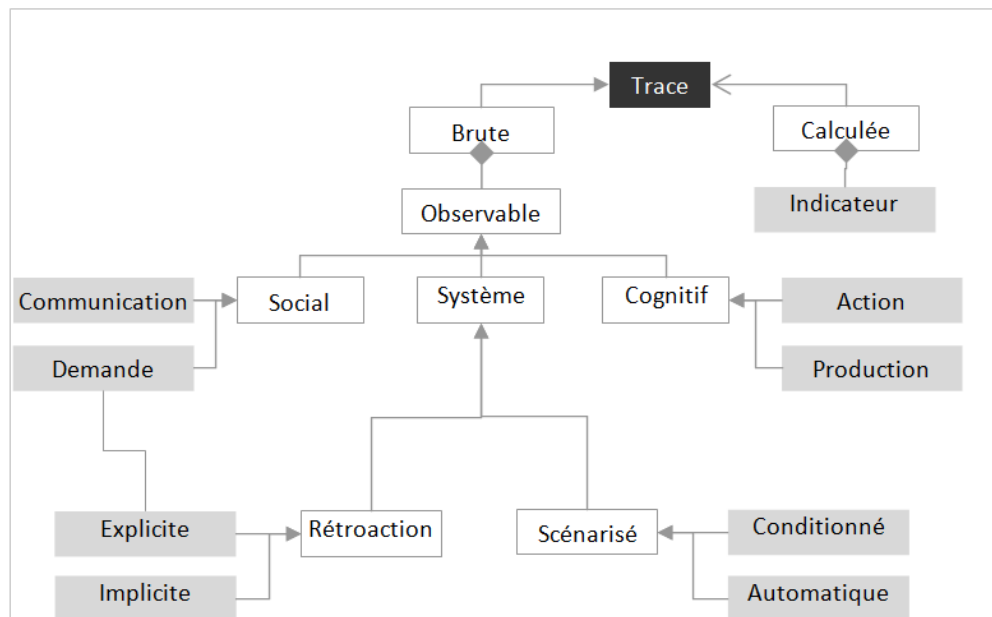


FIGURE 5.2 – Modèle de trace pour simulateur

- Observable : appelé aussi observé est une classe principale englobant tout type d'observé.
- Cognitif : observé englobant tous les comportements cognitifs demandant une réflexion et un raisonnement théorique et/ou pratique. Il peut être une configuration de paramètres, une affectation ou un changement de valeur (production) comme il peut être un lancement d'action (action).
- Social : observé décrit les communications entre les apprenants dans un contexte collaboratif (Communication), et entre l'apprenant et le simulateur sous forme de demandes d'aide, conseils, ...etc. (demande). Les comportements sociaux sont toujours initiés par l'apprenant, le cas inverse n'est pas considéré ici.
- Système : le comportement d'un simulateur peut être réactif suite à une demande explicite ou implicite de l'apprenant (rétroaction). Une demande explicite est lancée directement par l'apprenant alors qu'une demande implicite, est une réaction jugée pertinente suite à un comportement donné de l'apprenant. Le simulateur peut être proactif et son comportement dans ce cas est préétabli (scénarisé). Il se déclenche dans ce dernier cas soit automatiquement suite à un évènement temporel ou autre (automatique). Son déclenchement peut être conditionné par des comportements précis de l'apprenant ou par la présence des états aussi précis du simulateur.

5.2.2 Spécialisation du modèle de trace pour le cas de CUSIM

Le modèle de trace présenté précédemment peut être utilisé pour décrire les traces générées par notre outil CUSIM. Ceci est possible si on instancie les observables en feuilles dans l'hierarchie du modèle générique présenté dans la Figure 5.3. La Figure suivante donne une spécialisation plus détaillée des comportements cognitifs chez l'apprenant pour le cas du simulateur CUSIM. Les observés générés par CUSIM sont de type cognitif tant qu'ils demandent une réflexion et un raisonnement théorique et/ou pratique. Un observé peut être de type production s'il s'agit d'une configuration de paramètres, une affectation, un changement de valeur ou de type action s'il s'agit d'un lancement d'action.

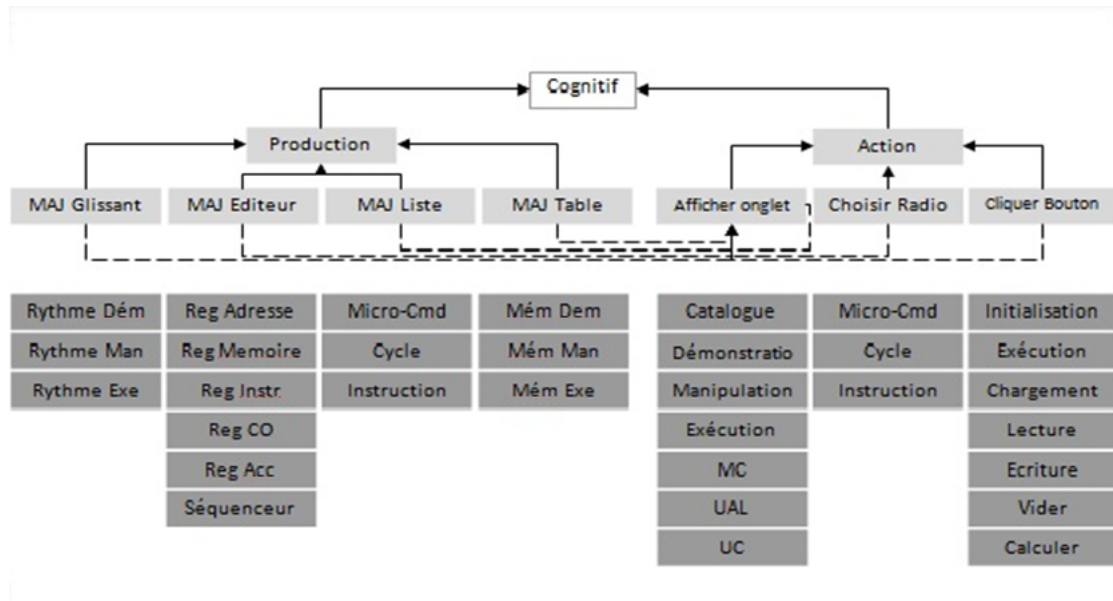


FIGURE 5.3 – Modèle de trace pour les observés générés par l'outil CUSIM

Les relations en lignes continues sont de type : « est une ».

Les relations en lignes pointillées sont de type : « nécessite ».

Ce modèle de trace est bien évidemment conçu sur la base d'une analyse de cet outil. Toutes les fonctions, les réactions et les productions sont discutées pour arriver à cette collection d'observés qui ont été affectés soigneusement aux feuilles du modèle précédent.

5.3 Approche proposée

Pour profiter des connaissances résidant dans le grand volume de traces générées par les EIAH sans se soucier du caractère technique incompréhensible par les enseignants-analystes, nous sommes obligés de proposer une version mieux adaptée et allégée à ces acteurs. Comme mentionné en introduction l'idée est de réécrire ces traces en utilisant un langage relativement simple basé sur ce que nous appelons *pattern*. Nous définissons un *pattern* dans notre contexte comme étant un morceau récurrent et utile de trace appelé aussi motif composé d'un ou plusieurs observés. Une suite utile d'observés est une suite pédagogiquement explicable porteuse de sémantique. L'approche que nous présenterons va consister à ressortir ces motifs à partir d'une trace brute pour former une trace réécrite beaucoup mieux exploitable et ce en utilisant une méthode d'appariement.

Pour expliquer notre approche nous allons considérer CUSIM comme l'EIAH dont nous voudrions interpréter les traces pour aider l'enseignant utilisant cet outil à comprendre le comportement de ces apprenants. L'approche est un processus composé de trois étapes. Le processus utilise une base de patterns pour être appariés avec le contenu des traces effectives dans chaque itération. L'initialisation de cette base est la première étape de ce processus. La deuxième étape consiste à appairer cette base en utilisant une certaine mesure de similarité afin de ressortir les patterns utiles. Pour rendre notre base évolutive et dynamique nous utilisons un mécanisme de recherche des séquences fréquentes qui s'applique sur le reste des traces traitées. Notre approche produit en sortie une trace réécrite de niveau sémantique plus élevé et prête à exploiter.

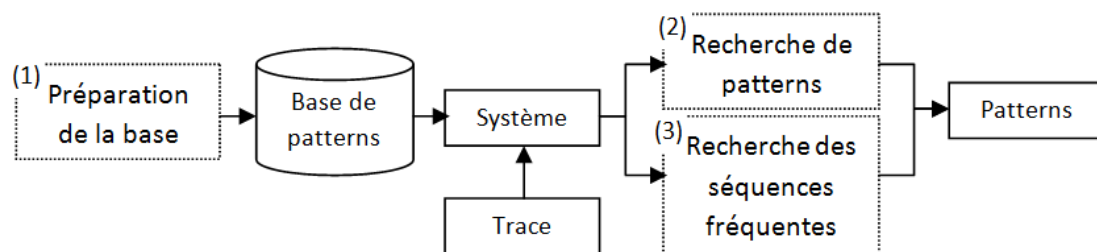


FIGURE 5.4 – Approche proposée

5.3.1 Préparation de la base

Nous entendons par préparation de la base l'insertion d'un ensemble initial de patterns. Ceci revient donc à imaginer selon l'EIAH en question le maximum de motifs d'observés. La base est construite de ces motifs avec pour chacun sa signification et la liste d'observés qui le compose. Pour augmenter encore plus le niveau d'abstraction de la trace nous proposons un deuxième niveau de patterns appelés les *super-patterns*. Ces derniers sont décrits dans la base de la même façon que les patterns mais en ras-

semblant des patterns au lieu des observés. Pour proposer un ensemble initial de patterns et de super-patterns avec leurs description (signification et liste d'observés/de patterns) respectives nous avons analysé soigneusement les différentes fonctionnalités de CUSIM, ses interfaces et ses moyens d'interaction. Concrètement cette base a été modélisée par une base de donnée composée de trois tables : une table *Obsel* représentant les éléments observés qui composent les patterns utiles, une table *Pattern* représentant les patterns ponctuelle qui eux-mêmes composent les super-patterns de la troisième table *Patterns*. Chaque super-pattern peut avoir un ou plusieurs patterns et chaque pattern peut avoir un ou plusieurs observés. Les super-patterns sont décrits par leur signification et la liste des patterns qui les composent. De la même manière, les patterns sont décrits par leur signification et la liste des observés qui les composent. Quatre informations sont plutôt utilisées pour décrire les observés : une information de temps (timestamp) ordonnant les observés dans le temps, le type de l'observé en considérant le modèle de la Figure 5.3, une description pour expliquer ce type et donner plus de détail sur lui et enfin le chemin parcouru par l'apprenant dans CUSIM pour réaliser une action ou déclencher un évènement.

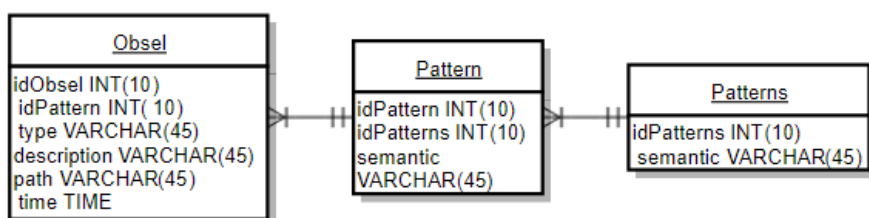


FIGURE 5.5 – Base de patterns

5.3.2 Recherche de patterns

Nous distinguons la reconnaissance de la recherche de patterns, la reconnaissance de patterns présuppose l'existence d'un modèle dont on va rechercher les occurrences parmi un ensemble de données. A l'inverse, la recherche de patterns (pattern mining) cherche à découvrir des motifs récurrents sans qu'aucun modèle ne soit suggéré [16]. A l'origine, le pattern mining a été utilisé à des fins commerciales pour analyser et développer des stratégies de marketing sur les achats des consommateurs [6]. Les éléments examinés peuvent être non ordonnés et dans ce cas seule l'appartenance d'un élément à l'une des catégories est significative. Le cas des éléments de l'ADN est un exemple sur les données totalement ordonnées où l'objectif est la recherche de séquences parfaitement identiques [7]. Dans d'autres applications, comme l'analyse des séquences d'actions d'un utilisateur, ou dans la détection d'erreurs en cascade dans les réseaux de télécommunication, les événements sont temporellement situés [69].

Le domaine de data mining a devenu aujourd'hui très utilisé est fructueux et ce pratiquement dans tous les champs scientifique, technique et autres. Nous voulons dans notre contexte profiter de ce succès et de ce potentiel pour interpréter nos traces en cherchant des motifs récurrents. La méthode que nous avons choisi est d'appliquer un appariement entre les patterns de la base et le contenu des traces brutes effectives. L'appariement utilise une mesure de similarité pour identifier les morceaux (patterns) utiles dans la trace. Nous appelons miettes le reste de la trace en termes d'observés après avoir enlever les parties utiles (les patterns détectés). Ces miettes ne correspondent à priori à aucun pattern de la base mais elles peuvent contenir des morceaux d'observés utiles qui ne figure encore pas dans la base. Pour profiter et ne pas jeter ces miettes d'observés nous avons proposer d'appliquer une autre technique de data mining qui est la recherche de motifs fréquents. Autrement dit nous cherchons les séquences d'observés fréquentes dans les miettes en appliquant un algorithme que nous avons adapté. Les éventuels motifs détectés résultants de l'algorithme seront proposés aux enseignants-analystes avant leur stockage dans la base de patterns et c'est avec ce mécanisme que la base s'évolue et s'accroître.

Pour pouvoir appliquer un algorithme d'appariement et une mesure de similarité, les éléments à appairer ou à en mesurer la similarité doivent être décrits et modélisés. De la même façon que dans la base de patterns les observés de la trace doivent aussi identifiés un par un et décrits par le type, la description et le chemin parcouru et ordonnés selon leurs timestamps. En s'inspirant des travaux de Sorlin [99] [109] nous avons opté pour une modélisation à base de graphs vu son caractère générique et fondement mathématiques solides. Une telle modélisation nous a été nécessaire pour appliquer la mesure de similarité que nous présenterons ultérieurement.

5.3.2.1 Identification et description des observés

Selon le type, le contenu et d'autres critères discutés dans le chapitre 2 le traitement automatique des traces peut être techniquement facile, délicat ou impossible. Pour identifier et extraire les observés d'une trace il suffit dans le cas de CUSIM de prendre chacune des lignes de son fichier texte car chaque observé occupe une ligne de texte. Les colonnes type, description, chemin sont bien délimitées dans CUSIM ce qui facilite aussi l'extraction de ces informations afin de décrire les observés. L'objectif ici est d'extraire le quadruplet (Temps, Type, Description, Chemin) de chaque ligne du fichier log, l'exemple suivant illustre cette opération tel que :

- Temps : représente le temps de production de l'observé.
- Type : décrit une production ou une action tout dépend du modèle de trace.
- Description : Explication de l'action faite par l'utilisateur.
- Chemin : le parcours des interfaces utilisé.

Exemple :

- T(00 :05, Action, Afficher l'onglet, Catalogue)

- T(00 :45, Action, Initialisation, Démonstration)
- T(03 :02, Action, Initialisation, Démonstration)
- T(03 :20, Action, Initialisation, Démonstration)
- T(03 :22, Action, Inst Suiv, Démonstration)
- T(03 :55, Action, Afficher l'onglet, Manipulation . Mémoire centrale)
- T(04 :14, Production, MAJ RA, Manipulation . Mémoire centrale)
- T(04 :25, Production, MAJ RM, Manipulation . Mémoire centrale)
- T(04 :36, Action, Ecriture mémoire, Manipulation . Mémoire centrale)

5.3.2.2 Appariement et mesure de similarité

Pour rendre l'appariement un processus générique et moins complexes les traces et les patterns devraient être représentés par un modèle unique. Nous allons adopter une représentation par les graphes et considérer dans la suite que la suite d'observés des patterns ou des traces brutes à traiter sont des sommets d'un graphe linéaire et le séquençement de ces observés représente les arêtes entre ces sommets. Dans notre graphe chaque sommet est caractérisé par un triplet (Type, Description, Chemin) et lié à un autre sommet par une arête suivant son ordre ou timestamp. Supposons E un ensemble de triplets (Type, Description, Chemin), $E = \{(O, D, C)^n, n \in \mathbb{N}\}$, le graphe P représentant les patterns et le graphe T représentant la trace brute sont définis comme suit : $(P, T) \in E \times E$.

L'appariement et la mesure de similarité que nous allons présenter sont une adaptation de ceux proposés par [31] et [99] en raison de sa simplicité et adéquation avec notre contexte. Dans [31] Champin dit que la similarité de deux graphes est calculée par rapport à un appariement des sommets de ces deux graphes et que l'appariement considéré est multivoque, i.e., chaque sommet d'un graphe peut être apparié à un ensemble -éventuellement vide- de sommets de l'autre graphe. Un appariement entre deux Graphes $P = \langle O1, D1, C1 \rangle$ et $T = \langle O2, D2, C2 \rangle$, est une relation : $m \in S_P \times S_T$ tel que S_P et S_T sont respectivement les sommets de P et T , contenant tout les couples de sommets des deux graphes.

A la différence de ce qu'il suppose [31] les arêtes de nos graphes ne sont pas étiquetées. La mesure de similarité entre P et T est ainsi adaptée comme suit :

$$sim_m(P, T) = \frac{f(P \sqcap_m T) - g(splits(m))}{f(\mathbb{C}_{S_P} \cup \mathbb{C}_{S_T})} \quad (5.1)$$

La formule précédente calcule la similarité entre une trace candidate T et un pattern P de la base, en mettant en correspondance leurs descriptions. $P \sqcap_m T$ représente l'ensemble des caractéristiques communes aux deux graphes par rapport à l'appariement considéré. \mathbb{C}_{S_P} et \mathbb{C}_{S_T} sont respectivement les caractéristiques (le triplet : $\langle O, D, C \rangle$ dans notre cas) des sommets de P et de T respectivement. f est une fonction qui pondère les caractéristiques des sommets et g est une fonction qui pondère les appariements redon-

dants ou splits. Ces fonctions définissent l'importance relative donnée aux descripteurs, les unes par rapport aux autres. Elles sont définies en fonction de l'application considérée : elles sont utilisées pour introduire les connaissances de similarité et les contraintes liées à l'application dans la mesure de similarité. En effet, les trois descripteurs O, D et C ne décrivent pas les observés avec la même importance. Une seule valeur du type O peut avoir plusieurs descriptions D et pas l'inverse. Le chemin à son tour n'identifie pas exactement un observé et on peut avec aussi plusieurs descriptions pour un même chemin. Le type O est par contre plus descriptif que le chemin parcouru car il donne au moins la catégorie de l'observé selon notre modèle de trace. Nous concluons de ces comparaisons que nos caractéristiques sont ordonnées du plus au moins important comme suit : D, O puis C. Au moment de l'implémentation nous pouvons définir f et g pour lesquelles respectent cet ordre d'importance.

Exemple

La figure suivante montre un exemple d'appariement d'un exemple de trace de CUSIM avec un pattern en mettant leurs caractéristiques en correspondance. Suivant l'appariement m les descripteurs communs sont encadrés en vert. L'application de la mesure de similarité précédente et si on considère que tous les poids des descripteurs sont égaux à 1, $Sim(Trace, Pattern) = 1$, l'appariement $m = \{(1', (3, 4, 5, 6, 7))(2', (8, 9, 10, 11, 12))(3', 13)\}$

Trace	Pattern
(1, Action, afficher l'onglet, catalogue)	(1', Production, MAJ RA, Manipulation .Mémoire centrale) ^a
(2, Action, Afficher l'onglet, Manipulation .Mémoire centrale)	(2', Production, MAJ RM, Manipulation .Mémoire centrale) ^a
(3, Production, MAJ RA, Manipulation .Mémoire centrale)	(3', Action, Écriture en mémoire, Manipulation .Mémoire centrale) ^a
(4, Production, MAJ RA, Manipulation .Mémoire centrale)	
(5, Production, MAJ RA, Manipulation .Mémoire centrale)	
(6, Production, MAJ RA, Manipulation .Mémoire centrale)	
(7, Production, MAJ RA, Manipulation .Mémoire centrale)	
(8, Production, MAJ RM, Manipulation .Mémoire centrale)	
(9, Production, MAJ RM, Manipulation .Mémoire centrale)	
(10, Production, MAJ RM, Manipulation .Mémoire centrale)	
(11, Production, MAJ RM, Manipulation .Mémoire centrale)	
(12, Production, MAJ RM, Manipulation .Mémoire centrale)	
(13, Action, Écriture en mémoire, Manipulation .Mémoire centrale)	

L'appariement $M = \{(3, 4, 5, 6, 7), 1'\}, \{(8, 9, 10, 11, 12), 2'\}, \{(13, 3')\}$

FIGURE 5.6 – Exemple d'un appariement pour les traces de CUSIM

5.3.2.3 Élimination des chevauchements

Le processus d'appariement produit une collection de patterns communs entre la base de patterns et les observés des traces effectives. Nous disons dans ce cas que nous

avons détecté des patterns similaires à ceux de la base dans notre trace. Ces patterns détectés peuvent se chevaucher sur la trace et nous obligent à faire le choix. En basant aussi sur notre mesure de similarité pour choisir entre le pattern à garder et celui qu'on ignore. Le pattern ayant une valeur de similarité supérieure est choisi ici.

L'exemple suivant illustre l'élimination de chevauchement entre deux patterns pattern1 et pattern2 en recourant à leurs mesures de similarité. Le chevauchement apparaît entre les lignes (3,4) du pattern1 et (1,2) du pattern 2. Après le calcul des similarités, on trouve le pattern 1 avec une similarité = 1 et pattern 2 avec une similarité = 0.75, le pattern 1 est choisi car sa mesure de similarité est supérieure.

TABLE 5.1 – Chevauchement des patterns

Extrait d'une trace CUSIM			
Timestamp	Activité	Sous Activité	Action
17 :29 :04	Manipulation	Mémoire centrale	MAJ Accumulateur
17 :29 :06	Manipulation	Mémoire centrale	MAJ opérande
17 :29 :10	Manipulation	Mémoire centrale	Choix opération
17 :29 :13	Manipulation	Mémoire centrale	Calcul
17 :29 :17	Manipulation	Unité de commande	Choix micro Cmd
17 :29 :22	Manipulation	Unité Arith/Log	Initialisation
Pattern 1			
Type	Description	Chemin	
Production	MAJ Accumulateur	Manipulation.Mémoire centrale	
Production	MAJ Opérande	Manipulation.Mémoire centrale	
Production	Choix opération	Manipulation.Mémoire centrale	
Action	Calcul	Manipulation.Mémoire centrale	
Pattern 2			
Type	Description	Chemin	
Production	MAJ Accumulateur	Manipulation.Mémoire centrale	
Production	MAJ Opérande	Manipulation.Mémoire centrale	
Production	Choix opération	Manipulation.Mémoire centrale	
Action	Calcul	Manipulation.Mémoire centrale	

5.3.3 Recherche des séquences fréquentes

La recherche de motifs fréquents permet de dégager des séquences ou épisodes d'actions dont le nombre élevé d'occurrences peut être porteur d'information. Les observés non reconnus en tant que membres dans des patterns détectés ne doivent pas être ignorés car ils peuvent former aussi des patterns utiles non détectés. La recherche de motifs récurrents dans un ensemble à caractère linéaire ou pas est un problème très

connu en data mining. Nous avons choisi dans notre contexte de nous appuyer sur l'algorithme SPADE [108] avec une légère adaptation et le travail de Cheype [32].

Dans cet algorithme l'extraction des motifs fréquents commence par la recherche des motifs de longueur 1, ensuite les motifs fréquents sont enregistrés et combinés entre eux pour former des motifs candidats de longueur supérieure. Les motifs non fréquents en 1 sont éliminés, et par conséquent aucun de leur super-motif n'est considéré. La fréquence des motifs candidats est testée pour constituer un nouvel ensemble de motifs fréquents et l'algorithme continue tant que de nouveaux candidats peuvent être formés.

Plus formellement cet algorithme s'appuie sur la notion de sous-séquence et de support. Une séquence $\alpha = (a_1, \dots, a_n)$ est une sous séquence de $\beta = (b_1, \dots, b_n)$ noté $\alpha \prec \beta$ s'il existe des entiers $i_1 < \dots < i_n$ tel que $a_i = b_{i_j}$. Le support d'une séquence α dans une trace T noté fr est le pourcentage de son apparition par rapport au nombre total des séquences s de même taille : $fr(\alpha, T) = |\alpha \in T| / |s \in T|, |\alpha| = |s|$. Une séquence α est alors dite fréquente dans une trace T quand $fr(\alpha, T) \geq SupportMin$ (choisi par l'analyste ou après un jeu d'essai).

L'algorithme SPADE adapté commence par déterminer les séquences de taille 1 qui sont fréquentes en calculant leur support et en ne gardant seulement que celles qui ont un support supérieur à SupportMin. Ensuite, pour déterminer les séquences de taille n , on sélectionne les candidats possibles de telle sorte que ces probables séquences de taille n fréquentes soient composées par n sous-séquences appartenant à l'ensemble des séquences de taille $n - 1$. On calcule ensuite leur support et on ne retient parmi les candidats que les séquences dont le support est supérieur à SupportMin.

Exemple : Soit la trace T définie comme suit :

$$T = \{ o3 \rightarrow o4 \rightarrow o1 \rightarrow o2 \rightarrow o1 \rightarrow o2 \rightarrow o5 \rightarrow o4 \rightarrow o3 \rightarrow o4 \\ \rightarrow o5 \rightarrow o1 \rightarrow o2 \rightarrow o5 \rightarrow o4 \rightarrow o2 \rightarrow o6 \rightarrow o1 \rightarrow o2 \rightarrow o5 \\ \rightarrow o2 \rightarrow o6 \rightarrow o4 \rightarrow o1 \rightarrow o2 \rightarrow o5 \rightarrow o4 \rightarrow o2 \rightarrow o6 \rightarrow o7 \},$$

$$SupportMin = 0.1.$$

Notre algorithme produit dans ses quatre itérations les résultats suivants :

- Itération 1 (séquences fréquentes de taille 1) :
 $o1(fr = 0.2), o2(fr = 0.17), o4(fr = 0.17), o5(fr = 0.17), o6(fr = 0.1)$.
- Itération 2 (séquences fréquentes de taille 2) :
 $o1 - o2(fr = 0.17), o2 - o5(fr = 0.13), o5 - o4(fr = 0.1), o2 - o6(fr = 0.1)$.
- Itération 3 (séquences fréquentes de taille 3) :
 $o1 - o2 - o5(fr = 0.1), o2 - o5 - o4(fr = 0.1)$.
- Itération 4 (séquences fréquentes de taille 4) :
 $o1 - o2 - o5 - o4(fr = 0.1)$.

5.4 LogAnalyser, un système pour la recherche de patterns dans les fichiers Log

Nous présentons ci-après un prototype réalisé dans le cadre d'un projet de Master réalisé par [36] concrétisant cette approche. Il a pour rôle de détecter les patterns dans les traces brutes ainsi que la détection de nouvelles séquences d'observés fréquentes et utiles.

Le système baptisé LogAnalyser est basé sur l'approche développée pour la recherche de patterns. Il permet à l'analyste-enseignant d'établir des déductions sur l'activité des apprenants. Il lui propose un certain nombre d'activités telles que gérer la base de patterns, ajouter et supprimer des patterns, consulter la liste des séquences fréquentes. Il peut aussi visualiser l'interprétation des traces d'un apprenant.

La figure suivante montre deux fonctionnalités essentielles de LogAnalyser, la visualisation des patterns détectés (à droite) et la recherche des séquences fréquentes (à gauche).

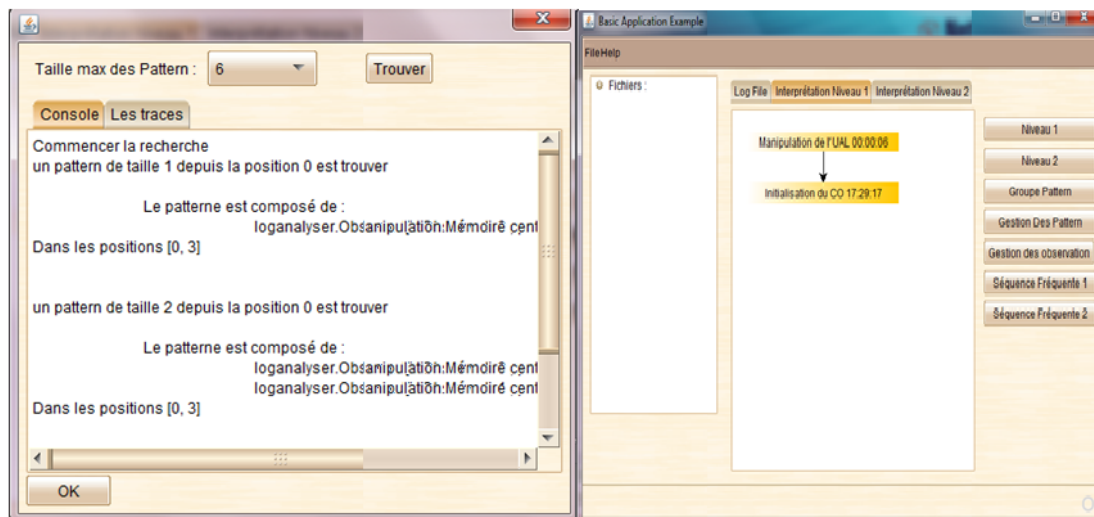


FIGURE 5.7 – Visualisation des patterns et recherche des séquences fréquentes dans LogAnalyser

5.5 Conclusion

Nous avons présenté dans ce chapitre notre proposition ainsi que l'environnement LogAnalyser conçu principalement pour la recherche de patterns dans des traces brutes d'interaction. L'approche consiste à chercher les patterns en utilisant un appariement entre la base de patterns et la trace brute en appliquant une mesure de similarité que

nous avons adapté pour les traces. Comme il se peut qu'il reste des séquences d'observés qui ne correspondent à aucun pattern dans la base. Pour palier à ce problème et profiter de ces séquences éventuellement utiles nous les avons enregistré et nous les avons fouiller afin de détecter les éventuelles séquences d'observés fréquentes. Nous avons présenté brièvement l'outil LogAnalyser qu'était le fruit de notre travail et à travers lequel nous avons pu tester et avec succès nos propositions, choix et adaptation à l'aide des interfaces graphique très simple à utiliser.

6

Conclusion

Plan du chapitre

4.1	Introduction	50
4.2	Processus Collecte et importation de traces	51
4.2.1	Génération du collecteur	51
4.2.2	Utilisation du collecteur	56
4.3	L'outil xCollector	57
4.3.1	Présentation	57
4.3.2	Exemple d'utilisation	58
4.4	Expérimentation	58
4.5	Conclusion	60

Nous concluons cette thèse par la discussion de nos contributions en se référant aux questions de recherche énoncées dans l'introduction. Les champs de l'assistance des utilisateurs dans les environnements informatiques et l'exploitation des traces informatiques d'utilisation ouvrent plusieurs possibilités de recherches supplémentaires. Quelques perspectives et travaux futurs possibles sont présentés en fin de cette conclusion.

6.1 Synthèse

Notre objectif principal dans cette thèse était de proposer une solution au problème de contrôle pédagogique dans les environnements informatiques pour l'apprentissage humain. Pour atteindre cet objectif, nous avons été amené à répondre à d'autres questions qui s'impliquent. Vu que nous avons affirmé le principe disant que la bonne compréhension de l'activité d'un apprenant conduit systématiquement à la bonne réaction dans un système informatique bien conçu (basé sur des règles ou autre mécanisme d'inférence), dans ce cas, la question est devenue « comment pouvoir comprendre l'activité de l'apprenant pour mieux lui donner de l'assistance pertinente ? » Les traces d'interaction étaient notre choix en tant que support porteur d'information très efficace et la question s'est transformée en la suivante : comment utiliser les traces d'interaction pour comprendre l'activité de l'apprenant ? La réponse à cette question faisait le sujet de notre troisième investigation présentée dans le quatrième chapitre. Cette question a été aussi traitée dans le troisième chapitre mais avec une autre vision. Sachant que d'une part, la réponse à cette question nécessite l'analyse des traces afin de les interpréter et de les comprendre, et que d'autre part, les systèmes à base de trace proposent entre autres des possibilités pareilles de traitement de traces (transformation, visualisation, indicateurs, interrogation), notre idée consiste à se servir de ces systèmes et à profiter ainsi de ces fonctionnalités. Cette piste nous a guidé à explorer un nouvel axe de recherche lié à la façon d'importer les traces dans ces systèmes. D'où la question de recherche suivante : comment importer des traces d'interaction de systèmes divers adoptant des formats et des contenus différents dans un système à base de trace utilisant un format unique afin de profiter de ses possibilités de traitement de traces ? La réponse à cette question faisait le sujet de notre deuxième investigation expliquée dans le troisième chapitre. Avant de commencer à chercher une solution au problème de collecte de traces, il était obligatoire de connaître au plus proche la nature des traces dans les systèmes informatiques afin de pouvoir proposer un système générique d'importation et de collecte. Ce besoin était justement l'objectif de l'étude sur les systèmes tracés et leurs traces, que nous avons mené et qui était notre première investigation de recherche dans ce manuscrit.

Dans l'intention d'avoir une idée assez claire sur le format, la structure, le type et le contenu des traces d'interaction générées par les systèmes tracés, nous avons voulu faire une étude d'analyse pour assurer que nos choix soient les plus génériques et les

plus standards possibles. En fouillant dans la littérature des EIAHs et des systèmes tracés en générales, nous avons pu choisir un nombre de logiciels devenus par la suite la matière de cette étude. Notre contribution a abouti à une brève explication sur chaque logiciel traité avec éventuellement des exemples de captures d'écran. Après consultation et analyse manuelle d'un ou de plusieurs exemples de traces de logiciel, une explication a été fournie. Cette explication s'intéressait aux formats de traces utilisés à savoir : texte, XML, base de données, enregistrement audio, vidéo, etc. Elle s'intéressait aussi au contenu des traces et au type d'informations enregistrées telles que les informations de profil, les productions saisies par l'utilisateur, les informations de communication, etc. Dans le cas des traces non structurées, des informations plus profondes sont fournies expliquant la structure de ces traces tels que les séparateurs utilisés et l'organisation interne.

La question que nous avons soulevé sur la façon d'importer des traces d'interaction de systèmes variés dans un système de gestion de bases de traces, se reformule dans la littérature en s'interrogeant sur la façon de faire la collecte de traces de systèmes variés pour un SBT. Pour répondre à la question nous avons proposé un système générique de collecte capable de collecter des traces de trois formats les plus fréquents en vue d'une importation dans KTBS. Le choix de ces trois formats s'appuyait sur l'étude des traces existantes que nous avons déjà réalisé (Chapitre 03). La généralité de notre système est garantie en proposant (général) un collecteur spécifique aux traces présentées. Les collecteurs sont générés après un processus d'apprentissage à base d'exemples. Plus précisément, pour qu'un collecteur choisisse automatiquement les morceaux d'informations intéressantes à collecter, il lui faut de l'apprentissage. Nous entendons par le terme à base d'exemple : prendre le ou les cas de traces qu'on lui présente et sur lesquelles on lui montre les morceaux d'informations intéressantes parmi celles non utiles. Afin d'imiter ce processus d'apprentissage et se comporter intelligemment devant un exemple de trace donné, nous avons muni notre système générateur d'une technologie XML très répondeuse appelée XPath. Cette dernière a la possibilité de repérer d'une façon souple des endroits spécifiques dans un contenu XML, d'où notre choix de passage par ce format de trace comme format intermédiaire. En revanche, notre système de collecte donne aussi la possibilité à ses utilisateurs de concevoir leur modèle de trace et de faire une sorte de liens nécessaires à la génération des collecteurs. Ces liens sémantiques entre les morceaux utiles de la trace et les éléments du modèle résument le rôle et l'importance de la collecte et du modèle de trace. Notre outil xCollecteur a été présenté comme étant la concrétisation de ce processus de collecte et plusieurs tests portés sur l'utilisabilité et l'efficacité de cet outil ont été faits chose qui valide ainsi les choix, les technologies et l'approche proposée.

La question que nous avons posée au sujet de la compréhension et l'interprétation des traces de manière à fournir de l'assistance à l'utilisateur, a été abordée et traitée par notre approche d'interprétation de traces. Cette approche est basée sur le concept d'appariement structurel appliqué avec une mesure de similarité entre les traces. L'approche consiste à chercher des patterns d'observés de trace en utilisant un appariement

entre une base de patterns et une trace brute. Ceci nous a incité à préparer une base de patterns de manière dynamique. La dynamique de la base signifie qu'elle s'enrichit à chaque cycle d'appariement suite à la détection de nouveaux patterns. Cependant, après que l'algorithme d'appariement instrumenté de notre mesure de similarité cherche et détecte tous les patterns qui se calquent ou se rapprochent à des patterns stockés dans la base, des séquences d'observés qui ne correspondent à aucun pattern se découlent. Ces portions d'observés peuvent être utiles et porteuses de sens, raison pour laquelle nous avons préféré de les garder. Ces observés sont possiblement utiles dans le sens qu'ils portent une sémantique propre et une structure stable même s'ils ne figurent pas dans la base de patterns en tant que membres d'un pattern propre. Pour traiter cette position, nous avons proposé d'enregistrer ces séquences d'observés dans une base spéciale et d'y chercher les éventuelles séquences d'observés fréquentes. Les séquences fréquentes résultantes sont ensuite présentées aux enseignants, celles jugées ayant une sémantique propre seront rajoutées en tant que nouveau pattern dans la base, les autres seront ignorées. LogAnalyser est l'outil réalisé pour tester et valider cette approche, il permet entre autre de visualiser l'interprétation des traces brutes d'une manière graphique. En plus de la recherche de patterns dans des traces brutes, le but pour lequel il est principalement conçu, l'outil LogAnalyser implémente aussi l'algorithme de recherche de patterns fréquents que nous avons expliqué auparavant.

6.2 Perspectives et travaux futures

Durant et même après la réalisation de ce modeste travail de recherche, plusieurs idées passent par l'esprit. Ces dernières peuvent devenir des pistes à explorer dans le futur. Nous voulons dans cette section donner celles les plus attirantes pour aider les chercheurs à entamer ou améliorer d'autres travaux de recherche dans le domaine de l'apprentissage.

Bien que l'étude que nous avons menée sur la nature des traces dans les EIAHs était relativement large et variée, elle nécessite tout de même une mise à jour. Un travail pareil actuellement pourrait être plus bénéfique et influent. Plusieurs logiciels éducatifs intéressants ont vu le jour après notre étude et d'autres n'ont plus d'existence pratiquement sur terrain. Maître à niveau cette étude est donc une nécessité pour donner de la valeur et de la crédibilité à une telle étude.

Même si notre travail de collecte et d'importation de traces traite les trois formats de traces les plus fréquents selon notre étude, ça sera plus intéressant et complémentaire de penser à collecter des traces d'autres formats. Un tel effort va donner plus de généralité et de caractère standard à notre travail. Plusieurs possibilités de repérage sont envisageables en plus de ce que nous avons proposé dans le processus de collecte. La complexité des contenus et des structures de traces nous obligent à penser davantage aux possibilités de repérage pour une identification plus précise des portions d'information utiles dans ces traces. Les traces collectées par notre système sont écrites en

RDF-Turtle, un des formats utilisé dans KTBS. Dans un travail futur, Il serai intéressant de proposer à l'utilisateur de choisir une représentation des traces collectées avec le format KTBS qui lui convient avant d'importer ses traces dans kTBS.

Notre approche d'interprétation de trace basée sur la recherche de patterns en s'appuyant sur un appariement structurel mérite aussi d'être développée et testée. En effet, notre choix d'utiliser une méthode structurelle basée sur la théorie des graphes et le choix d'une mesure telle que celle de Sorlin, sont des propositions parmi beaucoup d'autres qui peuvent donner des résultats similaires ou encore mieux. Reste donc à essayer d'autres pistes telles que les méthodes statistiques qui ont prouvé leurs efficacité et succès dans ce domaine comme dans d'autres.

Dans ce mémoire nous avons signalé à plusieurs reprise que nous somme pour le principe considérant qu'une bonne assistance est le résultat d'une bonne compréhension de ce qui se passe auprès de l'apprenant. Selon ce principe, la réaction du système assistant et donc le choix de l'action à faire devant une situation donnée, n'est qu'un processus automatique. Par contre, cette idée n'exclue pas qu'il y ait du travail à faire pour assurer cette réaction de la façon la plus efficace possible. Notre proposition a donc besoin d'être couplée avec ce travail de réaction. En perspective, nous pouvons parler ainsi d'un travail qui tourne autour des systèmes réactifs intelligents exploitant la description de la situation d'apprentissage (à partir de l'interprétation des traces tel que celui que nous avons proposé) pour donner des réactions pertinentes dépendantes aussi du profil de l'utilisateur.

Bibliographie

- [1] Université Claude Bernard Lyon 1. Logiciel de sensibilisation au tri sélectif et au recyclage. <https://perso.liris.cnrs.fr/stephanie.jean-daubias/projets/logiciel-TRI.html>, 2007. 41
- [2] Université Claude Bernard Lyon 1. Logiciel d'entraînement aux tables. <https://perso.liris.cnrs.fr/stephanie.jean-daubias/projets/logiciel-TablesAuTresor.html>, 2009. 43
- [3] Université Claude Bernard Lyon 1. elycee. <http://www.elycee.com/>, 2018. 42
- [4] Mari Abe and Masahiro Hori. A visual approach to authoring xpath expressions. *Markup languages theory and practice*, 3(2) :191–212, 2001. 54
- [5] Mari Abe and Masahiro Hori. Robust pointing by xpath language : Authoring support and empirical evaluation. In *Applications and the Internet, 2003. Proceedings. 2003 Symposium*, pages 156–165. IEEE, 2003. 54
- [6] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. In *Acm sigmod record*, volume 22, pages 207–216. ACM, 1993. 70
- [7] Rakesh Agrawal and Ramakrishnan Srikant. Mining sequential patterns. In *Data Engineering, 1995. Proceedings of the Eleventh International Conference on*, pages 3–14. IEEE, 1995. 70
- [8] Nikolaos Avouris, Vassilis Komis, Meletis Margaritis, and Georgios Fiotakis. An environment for studying collaborative learning activities. *Educational Technology & Society*, 7(2) :34–41, 2004. 40
- [9] Pierre-Loup Barazzutti, Amélie Cordier, and Béatrice Fuchs. Transmute : un outil interactif pour assister l'extraction de connaissances à partir de traces. In *EGC*, pages 463–468, 2016. 64
- [10] Thomas Beauvisage. Sémantique des parcours des utilisateurs sur le web. *PhD in Sciences du Langage, University of Paris X, Nanterre*, 2004. 20
- [11] Lawrence Bergman, Vittorio Castelli, Tessa Lau, and Daniel Oblinger. Docwizards : a system for authoring follow-me documentation wizards. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 191–200. ACM, 2005. 18

- [12] Mohamed Besnaci. Interprétation des traces d'interaction dans un eiah par une recherche de patterns d'activité. *RJC EIAH'2012*, page 15. 66
- [13] Mohamed Besnaci, Tahar Bensebaa, Nathalie Guin, and Pierre-Antoine Champin. Knowledge acquisition for importing existing traces to a trace base management system. *Journal of Information & Knowledge Management*, 17(04) :1850041, 2018. 51, 54
- [14] Mohamed Besnaci, Nathalie Guin, and Pierre-Antoine Champin. Importation de traces existantes dans un système de gestion de bases de traces, 2014. 57
- [15] Mohamed Besnaci, Nathalie Guin, and Pierre-Antoine Champin. Acquisition de connaissances pour importer des traces existantes dans un système de gestion de bases de traces. In *IC2015*, 2015. 51
- [16] Marie-Laure Betbeder, Régis Tissot, and Christophe Reffay. Recherche de patterns dans un corpus d'actions multimodales. In *Environnement Informatique pour l'Apprentissage Humain*, pages 533–544. Institut National de la Recherche Pédagogique-Association des Technologies de l'Information pour l'Education et la Formation, 2007. 65, 70
- [17] Mireille Betrancourt, Nicolas Guichon, and Yannick Prié. Assessing the use of a trace-based synchronous tool for distant language tutoring. In *Computer-Supported Collaborative Learning*, volume 1, pages 478–485, 2011. 25
- [18] Jeffrey P Bigham, Tessa Lau, and Jeffrey Nichols. Trailblazer : enabling blind users to blaze trails through the web. In *Proceedings of the 14th international conference on Intelligent user interfaces*, pages 177–186. ACM, 2009. 18
- [19] Denis Bouhineau. Edba : Exercises data base about al. https://edba.liglab.fr/index_EDBA_Full.html, 2012. 34
- [20] Denis Bouhineau, Vanda Luengo, Nadine Mandran, Michael Ortega, Claire Wajeman, et al. Open platform to model and capture experimental data in technology enhanced learning systems. In *Workshop Data Analysis and Interpretation for Learning Environments*, 2013. 50
- [21] Nabila Bousbia and Jean-Marc Labat. Perception de l'activité de l'apprenant dans un environnement de formation. In *Environnements Informatiques pour l'Apprentissage Humain (EIAH 2007)*, pages 233–238. INRP, 2007. 64
- [22] Willem-Paul Brinkman, Philip Gray, and Karen Renaud. Computer-assisted recording, pre-processing and analysis of user interaction data. In *Proc. HCI*, volume 2006, pages 273–275, 2006. 20
- [23] Robin Burke. Hybrid recommender systems : Survey and experiments. *User modeling and user-adapted interaction*, 12(4) :331–370, 2002. 12
- [24] Baptiste Cablé, Nathalie Guin, and Marie Lefevre. An authoring tool for semi-automatic generation of self-assessment exercises. In *International Conference on Artificial Intelligence in Education*, pages 679–682. Springer, 2013. 59

- [25] Antonio Capobianco and Noëlle Carbonell. Contextual online help : elicitation of human experts' strategies. In *Proceedings of HCI*, volume 1, pages 824–828. Citeseer, 2001. 12
- [26] Hamid Chaachoua, Marie-Caroline Croset, Denis Bouhineau, Marilena Bittar, Jean-François Nicaud, et al. Description et exploitations des traces du logiciel d'algèbre aplusix. *Revue des Sciences et Technologies de l'Information et de la Communication pour l'Education et la Formation (STICEF)*, 14, 2007. 34, 50
- [27] Hamid Chaachoua, Jean-François Nicaud, Alain Bronner, and Denis Bouhineau. Aplusix, a learning environment for algebra, actual use and benefits. In *ICME 10 : 10th International Congress on Mathematical Education, July 4-11, 2004*, page 8, 2004. 33, 59
- [28] Pierre-Antoine Champin, Amélie Cordier, E Lavoué, Marie Lefevre, and Alain Mille. Traces, assistance and communities, a review kolflow project-task 4-state of the art (d4. 1). Technical report, Research Report RR-LIRIS-2012-006, LIRIS,(April 2012), 2012. 19, 25
- [29] Pierre-Antoine Champin, Amélie Cordier, Élise Lavoué, Marie Lefevre, and Hala Skaf-Molli. User assistance for collaborative knowledge construction. In *Proceedings of the 21st International Conference on World Wide Web*, pages 1065–1074. ACM, 2012. 25
- [30] Pierre-Antoine Champin, Alain Mille, and Yannick Prié. Vers des traces numériques comme objets informatiques de premier niveau : une approche par les traces modélisées. *Intellectica*, 1 :59, 2013. 50
- [31] Pierre-Antoine Champin and Christine Solnon. Measuring the similarity of labeled graphs. In *International Conference on Case-Based Reasoning*, pages 80–95. Springer, 2003. 72
- [32] Adrien Cheype. Recherche de motifs séquentiels pour guider l'interprétation des traces d'apprentissage dans un eiah. *Actes de RJC-EIAH*, 6, 2006. 65, 75
- [33] Christophe Choquet, Sébastien Iksal, et al. Modélisation et construction de traces d'utilisation d'une activité d'apprentissage : une approche langage pour la réingénierie d'un eiah. *Revue des Sciences et Technologies de l'Information et de la Communication pour l'Education et la Formation (STICEF)*, 14, 2007. 51
- [34] Damien Cram, Béatrice Fuchs, Yannick Prié, and Alain Mille. An approach to user-centric context-aware assistance based on interaction traces. *MRC 2008, Modeling and Reasoning in Context*, 2008. 25
- [35] Damien Cram, Denis Jouvin, and Alain Mille. Visualisation interactive de traces et réflexivité : application à l'eiah collaboratif synchrone emédiathèque. *Sciences et Technologies de l'Information et de la Communication pour l'Éducation et la Formation*, 14(1) :491–530, 2007. 42, 64
- [36] Faiza Daoudi. *Interprétation des traces d'interaction d'un EIAH*. PhD thesis, Université BADJI MOKHTAT, Annaba, 2012. 76

- [37] Cédric d'Ham, Isabelle Girault, and Patricia Marzin. Des environnements numériques pour étayer l'investigation scientifique et la conception expérimentale : de copex-chimie à labbook. In *Skholê : cahiers de la recherche et du développement*, volume 18, pages 265–275, 2014. 38
- [38] Tarek Djouad. *Ingénierie des indicateurs d'activités à partir de traces modélisées pour un Environnement Informatique d'Apprentissage Humain*. PhD thesis, Université Claude Bernard-Lyon I, 2011. 64
- [39] Tarek Djouad, Lotfi Sofiane Settouti, Yannick Prié, Christophe Reffay, and Alain Mille. Un système à base de traces pour la modélisation et l'élaboration d'indicateurs d'activités éducatives individuelles et collectives. mise à l'épreuve sur moodle. *Mise à l'épreuve sur Moodle. Technique et Science Informatiques*, 2010. 23, 64
- [40] Ruihai Dong, Kevin McCarthy, Michael O'Mahony, Markus Schaal, and Barry Smyth. Towards an intelligent reviewer's assistant : recommending topics to help users to write better product reviews. In *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces*, pages 159–168. ACM, 2012. 18
- [41] Gregory Dyke, Kristine Lund, and Jean-Jacques Girardot. Tatiana, un environnement d'aide à l'analyse de traces d'interactions humaines. *Technique et Science Informatiques*, 29(10) :1179–1205, 2010. 64
- [42] Olivier Gapenne, Charles Lenay, and Dominique Boullier. Defining categories of the human/technology coupling : theoretical and methodological issues. In *Adjunct Proceedings of the 7th ERCIM Workshop on User Interface for All*, pages 197–198, 2002. 11
- [43] O Georgeon, B Mathern, A Mille, T Bellet, A Bonnard, N Henning, and JM TRÉ-MAUX. Abstract analysis of behavior and situation for mental representation assessment and cognitive activity modelling, 2007. 23
- [44] Olivier L Georgeon, Alain Mille, Thierry Bellet, Benoit Mathern, and Frank E Ritter. Supporting activity modelling from activity traces. *Expert Systems*, 29(3) :261–275, 2012. 24
- [45] Blandine Ginon, Stéphanie Jean-Daubias, and Pierre-Antoine Champin. Mise en place d'un système d'assistance personnalisée dans une application existante. In *IC-24èmes Journées francophones d'Ingénierie des Connaissances*, 2013. 25
- [46] Blandine Ginon, Stéphanie Jean-Daubias, and Pierre-Antoine Champin. Une typologie de l'assistance aux utilisateurs : exemple d'application aux eiah. Technical report, Technical Report RR-LIRIS-2013-007, LIRIS UMR 5205 CNRS/INSA de Lyon/Université Claude Bernard Lyon 1/Université Lumière Lyon 2/Ecole Centrale de Lyon.(Cited on page 4.), 2013. 11, 13, 14
- [47] Isabelle Girault and Cédric d'Ham. Scaffolding a complex task of experimental design in chemistry with a computer environment. *Journal of Science Education and technology*, 23(4) :514–526, 2014. 38

- [48] Nathalie Guin, Marie Lefevre, and Stéphanie Jean-Daubias. Personnalisation de l'apprentissage dans l'eiaa ambre-add. *Atelier " Personnalisation de l'apprentissage : quelles approches pour quels besoins ?"*, EIAH 2011, 2011. 39
- [49] Masahiro Hori, Mari Abe, and Kouichi Ono. Extensible framework of authoring tools for web document annotation. In *Proceedings of International Workshop on Semantic Web Foundations and Application Technologies (SWFAT)*, pages 1–8. Citeseer, 2003. 53, 54
- [50] Masahiro Hori, Mari Abe, and Kouichi Ono. Robustness of external annotation for web-page clipping : Empirical evaluation with evolving real-life web documents. In *Semannot 2003 : Knowledge Markup and Semantic Annotation*, Sanibel, Florida, USA, 2003. 54
- [51] Masahiro Hori, Kouichi Ono, Mari Abe, and Teruo Koyanagi. Generating transformational annotation for web document adaptation : tool support and empirical evaluation. *Web Semantics : Science, Services and Agents on the World Wide Web*, 2(1) :1–18, 2004. 54
- [52] Alison Hull and Benedict du Boulay. Motivational and metacognitive feedback in sql-tutor. *Computer Science Education*, 25(2) :238–256, 2015. 37
- [53] Stéphanie Jean-Daubias. *PÉPITE : un système d'assistance au diagnostic de compétences*. PhD thesis, Université du Maine, 2000. 41
- [54] Stéphanie Jean-Daubias. Tables au trésor, un logiciel d'entraînement aux tables mathématiques entièrement paramétrable. *Atelier " Personnalisation de l'apprentissage : quelles approches pour quels besoins ?"*, EIAH 2011, 2011. 43
- [55] Stéphanie Jean-Daubias. Tri, un logiciel de sensibilisation au tri sélectif et au recyclage s' appuyant sur des sessions personnalisables. *Atelier " Personnalisation de l'apprentissage : quelles approches pour quels besoins ?"*, EIAH'2011, 2011. 42
- [56] Min Ji, Christine Michel, Elise Lavoué, and Sébastien George. An architecture to combine activity traces and reporting traces to support self-regulation processes. In *Advanced Learning Technologies (ICALT), 2013 IEEE 13th International Conference on*, pages 87–91. IEEE, 2013. 50
- [57] Hyunmo Kang, Catherine Plaisant, and Ben Shneiderman. New approaches to help users get started with visual interfaces : multi-layered interfaces and integrated initial guidance. In *Proceedings of the 2003 annual national conference on Digital government research*, pages 1–6. Digital Government Society of North America, 2003. 18
- [58] Caitlin Kelleher and Randy Pausch. Stencils-based tutorials : design and evaluation. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 541–550. ACM, 2005. 18
- [59] Takatoshi Kitano, Keiichi Iguchi, and Kazuya Koyama. Generating robust xpathes for service customization. In *6th World Congress on Services (SERVICES-1)*, pages 166–167. IEEE, 2010. 53

- [60] Kenneth R Koedinger, Ryan Sjd Baker, Kyle Cunningham, Alida Skogsholm, Brett Leber, and John Stamper. A data repository for the edm community : The pslc datashop. *Handbook of educational data mining*, 43 :43–56, 2010. 31
- [61] Marek Kowalkiewicz, Tomasz Kaczmarek, and Witold Abramowicz. Myportal : robust extraction and aggregation of web content. In *Proceedings of the 32nd international conference on Very large data bases*, pages 1219–1222. VLDB Endowment, 2006. 53
- [62] Julien Laflaquière, Yannick Prié, and Alain Mille. Ingénierie des traces numériques d’interaction comme inscriptions de connaissances. In *19es Journées Francophones d’Ingénierie des Connaissances (IC 2008)*, pages 183–195, 2008. 20
- [63] Marie Lefèvre. *Processus unifié pour la personnalisation des activités pédagogiques : méta-modèle, modèles et outils*. PhD thesis, Université Claude Bernard-Lyon I, 2009. 31
- [64] David Leray and Jean-Paul Sansonnet. Acquisition de connaissances perceptives pour un agent assistant. In *18es Journées Francophones d’Ingénierie des Connaissances*, pages not-specified, 2007. 12
- [65] Henry Lieberman. Interfaces that give and take advice, 2001. 21
- [66] METAH LIG. Tp-elec. <http://tpelec.imag.fr/release/1.9.5/authentication.php>, 2008. 37
- [67] lip6. Superviseur project. <http://superviseur.lip6.fr/>, 2013. 59
- [68] Magnus S Magnusson. Discovering hidden time patterns in behavior : T-patterns and their detection. *Behavior Research Methods, Instruments, & Computers*, 32(1) :93–110, 2000. 21
- [69] Heikki Mannila, Hannu Toivonen, and A Inkeri Verkamo. Discovery of frequent episodes in event sequences. *Data mining and knowledge discovery*, 1(3) :259–289, 1997. 70
- [70] Benoît Mathern, Thierry Bellet, and Alain Mille. An iterative approach to develop a cognitive model of the driver for human centred design of its. In *European Conference on Human Centred Design for Intelligent Transport Systems, Proceedings of European Conference on Human Centred Design for Intelligent Transport Systems*, pages 85–95, 2010. 24
- [71] Madeth May, Sébastien George, and Patrick Prévôt. A closer look at tracking human and computer interactions in web-based communications. *Interactive Technology and Smart Education*, 5(3) :170–188, 2008. 50
- [72] Madeth May, Sébastien George, and Patrick Prévôt. Tracer, analyser et visualiser les activités de communications médiatisées des apprenants. In *Colloque JOCAIR 2008 (Journées Communication et Apprentissage Instrumentées en Réseau)*, pages p–251. Hermès Sciences, Lavoisier, 2008. 64
- [73] Agathe Merceron and Kalina Yacef. Educational data mining : a case study. In *AIED*, pages 467–474, 2005. 65

- [74] Laboratoire LIG METAH. Copex-chimie : Construire un protocole expérimental de dosage spectrophotométrique. <http://copex-chimie.imag.fr>, 2018. 38
- [75] Sandra Michelet, Vanda Luengo, Jean-Michel Adam, and Nadine Mandran. Diagnosing knowledge using learning activity traces generated by various problem-solving modalities. *International Journal of Learning Technology*, 9(1) :67–90, 2014. 37
- [76] Antonija Mitrović. Experiences in implementing constraint-based modeling in sql-tutor. In *International Conference on Intelligent Tutoring Systems*, pages 414–423. Springer, 1998. 36
- [77] Rejane Monod-Ansaldi, Eric Sanchez, Daniel Devallois, Thomas Abad, Pierre Benech, Anne Brondex, Isabelle Mazzella, Sandrine Miranda, Claudie Richet, and Celine Recurt. Sit down to table and confess who you are! design of an epistemic game for nutritional education at secondary school. In *European Conference on Games Based Learning*, page 401. Academic Conferences International Limited, 2013. 59
- [78] Sandra Nogry, Nathalie Guin, and Stéphanie Jean-Daubias. Ambre-add : An its to teach solving arithmetic word problems. *Technology, Instruction, Cognition and Learning*, 6 :53–61, 2008. 39, 59
- [79] Pittsburgh Science of Learning Center. Datashop@cmu. <https://pslcldata-shop.web.cmu.edu/>, 2018. 31
- [80] Magali Ollagnier-Beldame. *Traces d'interactions et processus cognitifs en activité conjointe : Le cas d'une co-rédaction médiée par un artefact numérique*. PhD thesis, Lyon 2, 2006. 21
- [81] F.Conil P.A Champin. ktbs : a kernel for trace-based systems. <https://kernel-for-trace-based-systems.readthedocs.io/en/latest/>, 2018. 50
- [82] Gilbert Paquette, François Pachet, and Sylvain Giroux. Épitalk, un outil générique pour la construction de systèmes conseillers. *Revue Sciences et Techniques Educatives*, 1(3) :305–336, 1994. 17
- [83] Gilbert Paquette and Pierre Tchounikine. Contribution à l'ingénierie des systèmes conseillers : une approche méthodologique fondée sur l'analyse du modèle de la tâche. *Sciences et Technologies de l'Information et de la Communication pour l'Éducation et la Formation*, 9(3) :409–435, 2002. 12
- [84] Iñaki Paz and Oscar Díaz. Providing resilient xpaths for external adaptation engines. In *Proceedings of the 21st ACM conference on Hypertext and hypermedia*, pages 67–76. ACM, 2010. 53
- [85] Jean-Philippe Pernin and Hélène Godinet. Aci “terrains, techniques, théories”. 42
- [86] Catherine Plaisant, Anne Rose, Gary Rubloff, Richard Salter, and Ben Shneiderman. The design of history mechanisms and their use in collaborative educational

- simulations. In *Proceedings of the 1999 conference on Computer support for collaborative learning*, page 44. International Society of the Learning Sciences, 1999. 21
- [87] Jörg Rech, Eric Ras, and Björn Decker. Intelligent assistance in german software development : A survey. *IEEE software*, 24(4), 2007. 15, 16
- [88] Bernhard Rieder. Pratiques informationnelles et analyse des traces numériques : de la représentation à l'intervention. *Études de communication. langages, information, médiations*, (35) :91–104, 2010. 64
- [89] Eric Sanchez, Shawn Young, and Caroline Jouneau-Sion. Classcraft : from gamification to ludicization of classroom management. *Education and Information Technologies*, 22(2) :497–513, 2017. 59
- [90] Eric Sanchez, Olivier Lefevre. Geonote : un environnement informatique d'aide au travail sur le terrain pour l'enseignement des sciences de la terre. In *Biennale de l'éducation*, Lyon, 2006. 50, 58, 59
- [91] Guus Schreiber and Yves Raimond. Rdf 1.1 primer. w3c working group note.(24 june 2014). W3C. *Recuperado a partir de <https://www.w3.org/TR/2014/NOTE-rdf11-primer-20140624>*, 2014. 20
- [92] Kay G Schulze, Robert N Shelby, Donald J Treacy, Mary C Wintersgill, Kurt Vanlehn, and Abigail Gertner. Andes : An intelligent tutor for classical physics. *Journal of Electronic Publishing*, 6(1), 2000. 35
- [93] Karim Sehaba. Système d'aide adaptatif à base de traces. *Revue internationale des technologies en pédagogie universitaire/International Journal of Technologies in Higher Education*, 9(3) :55–70, 2012. 25
- [94] Lemya Settouti, Nathalie Guin, Alain Mille, and Vanda Luengo. A trace-based learner modelling framework for technology-enhanced learning systems. In *Advanced Learning Technologies (ICALT), 2010 IEEE 10th International Conference on*, pages 73–77. IEEE, 2010. 23
- [95] Lotfi Settouti. *Systèmes à Base de Traces Modélisées : Modèles et Langages pour l'exploitation des traces d'Interactions*. PhD thesis, Lyon 1, 2011. xi, 21, 22, 23, 25
- [96] Lotfi Sofiane Settouti, Yannick Prié, Jean-Charles Marty, and Alain Mille. Vers des systèmes à base de traces modélisées pour les eiah. *Rapport de recherche RR-LIRIS-2007*, 16, 2007. 22
- [97] Jérôme Simonin and Noëlle Carbonell. Interfaces adaptatives adaptation dynamique à l'utilisateur courant. *arXiv preprint arXiv :0708.3742*, 2007. 13
- [98] Amy Soller, Alejandra Martínez, Patrick Jermann, and Martin Muehlenbrock. From mirroring to guiding : A review of state of the art technology for supporting collaborative learning. *International Journal of Artificial Intelligence in Education*, 15(4) :261–290, 2005. 21
- [99] Sébastien Sorlin, Olfa Sammoud, Christine Solnon, and Khaled Ghédira. Comparaison de l'optimisation par colonies de fourmis et d'une recherche tabouée réactive sur des problèmes d'appariement de graphes. 71, 72

- [100] Ben-Manson Toussaint, Vanda Luengo, Francis Jambon, and Jérôme Tonetti. Modeling perceptual-gestural knowledge for intelligent tutoring systems dedicated to ill-defined domains. *International Journal of Artificial Intelligence in Education (IJAIED)*, 20(3), 2010. 32
- [101] Ben-Manson Toussaint, Vanda Luengo, Lucile Vadcard, and Jérôme Tonetti. Apprentissage de la chirurgie orthopédique assisté par ordinateur : Le cas du système tutoriel intelligent teleos. *Field Actions Science Reports. The Journal of Field Actions*, (Special Issue 9), 2014. 32
- [102] Martin Dougiamas Curtin University. Moodle. <https://moodle.com/>, 2018. 44
- [103] Kurt VanLehn, Brett Van De Sande, Robert Shelby, and Sophia Gershman. The andes physics tutoring system : An experiment in freedom. In *Advances in intelligent tutoring systems*, pages 421–443. Springer, 2010. 35
- [104] W3C. Extensible markup language (xml). <http://www.w3.org/TR/xml/>, 2008. 52
- [105] W3C. Xml path language (xpath) version 2.0. <http://www.w3.org/TR/xpath20/>, 2010. 53
- [106] Etienne C Wenger and William M Snyder. Communities of practice : The organizational frontier. *Harvard business review*, 78(1) :139–146, 2000. 13
- [107] Tom Yeh, Tsung-Hsiang Chang, Bo Xie, Greg Walsh, Ivan Watkins, Krist Wongsuphasawat, Man Huang, Larry S Davis, and Benjamin B Bederson. Creating contextual help for guis using screenshots. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 145–154. ACM, 2011. 18
- [108] Mohammed J Zaki. Spade : An efficient algorithm for mining frequent sequences. *Machine learning*, 42(1-2) :31–60, 2001. 75
- [109] Stéphane Zampelli, Yves Deville, Christine Solnon, Sébastien Sorlin, and Pierre Dupont. Filtrage pour l’isomorphisme de sous-graphe. In *Troisièmes Journées Francophones de Programmation par Contraintes (JFPC07)*, 2007. 71
- [110] Raafat Zarka. *Trace-based reasoning for user assistance and recommendations*. PhD thesis, INSA de Lyon, 2013. 19
- [111] Raafat Zarka, Amélie Cordier, Françoise Corvaisier, and Alain Mille. Providing assistance by reusing episodes stored in traces : a case study with sap business objects explorer. *RàPC 2010*, page 91, 2010. 13