

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université De Guelma

Faculté de Mathématiques et de l'Informatique

Département d'informatique



Mémoire



En vue de l'obtention du diplôme de

Magister en Informatique

Option : Intelligence Artificielle (IA)

**Réutilisation des traces dans un
Environnement d'Apprentissage Humain**

Présenté par

Mme FELKAOUI SOUMIA

Jury :

Mr SERIDI Hamid	Prof. (Univ, Guelma)	Président
Mr LAFIFI Yacine	M.C (Univ, Guelma)	Examineur
Mr MOKHATI Farid	M.C (Univ, O.E.Bouaghi)	Examineur
Mme SERIDI Hassina	M.C (Univ, Annaba)	Directrice de mémoire

Année universitaire 2011-2012

Dédicaces

A ceux qui ont attendu avec patience les fruits de leur éducation : mes très chers parents. Que Dieu leur préserve longue vie pleine de santé et de joie.

A mon très cher mari.

A mes très chères sœurs Nadjoua et Meriem

A mes très chères frères Tarek et Zaki

A mes très chers neveux Hamada, Choib et le petit Moetaz Billah

A ma très chère nièce Maya (RYMMECE)

A la mémoire de ma belle mère YAMINA. C'est avec une grande douleur de ne plus l'avoir parmi nous

A mon bel père, mes beaux frères et mes belles Sœurs

A ma très chère sœur WARDA

A toutes mes amies

A toute la famille FELKAOUI et SLIMANI

Remerciements

Tout d'abord, Je tiens à exprimer ma profonde gratitude, mes respects et mes plus vifs remerciements à mon Directeur de recherche Dr SERIDI Hassina, pour avoir encadré et dirigé mes recherches. Je la remercie pour toute la confiance qu'elle a su me porter, et pour la patience et pour ses encouragements dont il a fait preuve à mon égard. Ses conseils et remarques constructives m'ont permis d'améliorer grandement la qualité de mes travaux et de ce mémoire.

Je tiens à exprimer ma profonde gratitude, mes respects et mes plus vifs remerciements à mon Co-encadreur Dr Yacine LAFIFI, pour sa patience, ses conseils, ses efforts, et ses encouragements dont il a fait preuve à mon égard. Qu'il trouve ici nos plus profondes reconnaissances.

Des remerciements qui n'ont pas de limite, je les adresse à Mme Ghalia MERZOUGUI et Dr. Bahloul (Univ. Batna), pour ses aides précieuses, ses bonté de cœur, ses encouragements généreux. Merci pour tout, Je suis reconnaissante envers vous.

Je ne pourrais jamais oublier le soutien et l'aide de mes proches pour m'avoir soutenu par leur présence dans les bons comme dans les mauvais moments : mes parents et qu'ils sachent que leurs présence à mes côtés a été précieuse et réconfortante.

Enfin, je souhaite associer à mes remerciements à mon mari qui m'a toujours encouragé, partagé mes doutes comme mes avancées. Ses précieuses remarques et idées m'ont encouragé dans mon travail. Merci pour ton soutien inconditionnel.

Sommaire

INTRODUCTION GENERALE	1
------------------------------------	----------

CHAPITRE I: LES TRACES DANS LES EIAH

1. INTRODUCTION	4
------------------------------	----------

2. TRACES DANS LES EIAH.....	5
-------------------------------------	----------

2.1. ENVIRONNEMENT D'APPRENTISSAGE HUMAIN (EIAH)	5
--	---

2.2. EXPLOITATION DES TRACES DANS LES EIAH	6
--	---

2.3. TYPES DE TRACES	7
----------------------------	---

2.4 CLASSIFICATION DES TRACES NUMERIQUES	8
--	---

2.4.1 CLASSIFICATION SELON LES INTERACTIONS ENTRE UTILISATEURS.....	8
---	---

2.4.2 CLASSIFICATION SELON LE CONTENU.....	9
--	---

2.5. TRAITEMENT DE TRACES NUMERIQUES.....	11
---	----

2.5.1 APPROCHES FONDEES SUR L'ANALYSE DES FICHIERS LOGS	11
---	----

2.5.2 APPROCHES FONDEES SUR DES MECANISMES D'OBSERVATION SPECIFIQUES A UN OUTIL D'APPRENTISSAGE PARTICULIER	13
--	----

2.5.3 APPROCHES FONDEES SUR DES MODELES GENERAUX	14
--	----

3. DEFINITIONS DES CONCEPTS UTILISÉS DANS CE MÉMOIRE.....	15
--	-----------

3.1. OBJET D'APPRENTISSAGE	15
----------------------------------	----

3.2. ACTIVITE PEDAGOGIQUE	15
---------------------------------	----

3.3. TRACE.....	16
-----------------	----

3.4. TRACE MODELISEE	16
----------------------------	----

• MODELE DE TRACE	16
-------------------------	----

3.5. SYSTEME A BASE DE TRACE (SBT).....	17
---	----

4. CONCLUSION	18
----------------------------	-----------

CHAPITRE II: L'UTILISATION DES TRACES DANS LES EIAH

1. INTRODUCTION	19
------------------------------	-----------

2. UTILISATION DES TRACES DANS LES EIAH	19
--	-----------

2.1 TRACES POUR L'ENSEIGNANT	20
------------------------------------	----

A- LE CALCUL D'INDICATEURS.....	21
---------------------------------	----

B- LES PARCOURS DES APPRENANTS	22
--------------------------------------	----

2.2 TRACES POUR L'APPRENANT	23
-----------------------------------	----

A. TRACE COMME MIROIR DE L'ACTIVITE DE L'APPRENANT.....	23
---	----

B. TRACE COMME FACILITATEUR DE L'ACTIVITE DE L'APPRENANT	23
2.3 TRACES POUR L'ENSEIGNANT-CONCEPTEUR D'ACTIVITES PEDAGOGIQUES.....	24
2.4 TRACES POUR LE CHERCHEUR	25
3. TYPES DES SYSTEMES QUI TRACENT LES INTERACTIONS UTILISATEUR- SYSTEME	25
3.1 SYSTEMES UTILISANT L'HISTOIRE INTERACTIONNELLE SANS LA PRESENTE AUX UTILISATEURS	25
3.2. SYSTEMES PRESENTANT UNE VISUALISATION DE L'HISTOIRE INTERACTIONNELLE DESTINE A L'ANALYSE DE LA SITUATION	26
3.3. SYSTEMES PRESENTANT UNE VISUALISATION DE L'HISTOIRE INTERACTIONNELLE POUR L'UTILISATEUR ET LUI PROPOSANT LA POSSIBILITE D'Y NAVIGUER.....	26
3.4. SYSTEMES PRESENTANT UNE VISUALISATION DE L'HISTOIRE INTERACTIONNELLE DESTINE A L'UTILISATEUR ET LUI PROPOSANT LA POSSIBILITE D'AGIR DESSUS	27
4. SYSTEMES A BASE DE TRACES MODELISEES	27
4.1. NOTION DE SYSTEMES A BASE DE TRACES MODELISEES	27
4.2 DIFFERENCES ENTRE LES SBC ET LES SBT	28
4.3. FONCTIONNEMENT DE SYSTEMES A BASE DE TRACES MODELISEES	28
5. MODELES ET REPRESENTATIONS CONSIDEREES POUR LES TRACES	30
5.1. LE MODELE DU LANGAGE UTL	30
5.2. LE MODELE DU LANGAGE UTL2	31
5.3. CLASSIFICATION DES TYPES DE TRACES DECRIVANT DES PARCOURS D'APPRENTISSAGE PROPOSEE PAR LE PROJET TRAILS	32
5.4. LE PROJET LISTEN	33
5.5. L'APPROCHE SYSTEME A BASE DE TRACES MODELISEES (SBT)	34
• LE MODELE DE TRACE	35
• LE SERVICE DE REQUETE (SBT KERNEL).....	35
• LE SYSTEME DE COLLECTE.....	36
• LE SYSTEME DE VISUALISATION	36
• LE SYSTEME DE GESTION DES TRACES	36
• AVANTAGE DES SBT	36
6. CONCLUSION	37
CHAPITRE III: LA REUTILISATION DES CONNAISSANCES	
1. INTRODUCTION	38
2. LA REUTILISATION.....	38

2.1 ORIGINES ET CONCEPTS	38
2.2. L'APPORT DE LA REUTILISATION EN DIFFERENTES DISCIPLINES	39
3. LA REUTILISATION DES CONNAISSANCES DANS LES EIAH	41
3.1. REPRESENTATION DE LA REUTILISATION	41
3.2. REUTILISER LES TRACES D'APPRENTISSAGE.....	42
A) POURQUOI REUTILISE-T-ON LES TRACES D'APPRENTISSAGE ?	42
B) POUR QUI LA REUTILISATION DES TRACES D'APPRENTISSAGE EST CIBLEE ?	42
C) COMMENT MEMORISER LES TRACES D'APPRENTISSAGES ?.....	43
4. LES TRAVAUX EXISTANTS CONCERNANT LA REUTILISATION	43
4.1. PROJETS DE REUTILISATION DANS LE DOMAINE DES RESSOURCES PEDAGOGIQUES	43
4.1.1. LE PROJET CDE	43
4.1.2. LE SYSTEME OLA	44
4.1.3. LE PROJET ARIADNE	44
4.1.4. LE PROJET SEMUSDI	44
4.1.5. SCORM	45
4.1.6. DISCUSSION	45
4.2. PROJETS SUR REUTILISATION DES TRACES D'APPRENTISSAGE.....	46
4.2.1. LIFELINE.....	46
4.2.2. L'EIAH SIMPLE	46
4.2.3. PADPRINTS	46
4.2.4. LE SYSTEME FOOTPRINTS.....	46
4.2.5. L'ENVIRONNEMENT DREW	47
4.2.6. MUSETTE.....	47
4.2.7. LE PROJET EPICEA	48
4.2.8. LE PROJET PIXED	49
4.2.9. LE SYSTEME ATER	50
4.2.10. DISCUSSION.....	52
5. CONCLUSION	53

CHAPITRE IV: LA CONCEPTION DU SYSTEME SRTn

1. INTRODUCTION	54
2. ARCHITECTURE DU SYSTEME SRTN	54
2.1. DEFINITIONS.....	54
A. TRACE PREMIERE	54

B. TRACE MODELISEE	55
C. TRACE DE-MODELISEE	55
D. EPISODE.....	55
E. MODELE DE TRACES	55
2.2 ARCHITECTURE GENERALE DU SYSTEME SRTN	55
2.3 ARCHITECTURE FONCTIONNELLE DU SYSTEME SRTN	56
3. PHASE DE PREPARATION	57
3.1. TYPE DE TRACES A COLLECTER PAR LE SYSTEME SRTN	57
3.2. COLLECTE ET FILTRAGE DE TRACES	59
A) LISTES DES TRACES A COLLECTER	60
B) POURQUOI UTILISER DES TRACES MODELISEES ?	60
3.3 PROCESSUS DE TRANSFORMATION.....	63
3.3.1 STRUCTURE GENERALE D'UN EPISODE	64
3.3.2. STRUCTURE FORMELLE D'UN EPISODE.....	65
3.4. INDEXATION.....	68
4. PHASE D'EXPLOITATION	69
4.1. MESURES DE SIMILARITE	70
4.1.1 CALCUL DE SIMILARITE ENTRE EPISODES BASE SUR LA TABLE D'INDEX.....	70
4.1.2 CALCUL DE SIMILARITE ENTRE EPISODES BASE SUR LES DISTANCES ENTRE SES ELEMENTS.....	71
4.2 VISUALISATION DES TRACES	74
5. MODEM DE TARACES	75
5.1 LA REUTILISATION DES TRACES D'APPRENTISSAGE COLLECTEES	75
5.2 LA VISUALISATION DE TRACES	75
5.3 LES AVANTAGES DU MODEM DE TRACES.....	76
6. CONCLUSION	76
CHAPITRE V: LA VALIDATION ET L'IMPLEMENTATION DU PROTOTYPE	
1. INTRODUCTION	78
2. EXEMPLE ILLUSTRATIF	78
2.1 COLLECTE ET TRANSFORMATION DE TRACES.....	78
2.1 EXTRACTION DES EPISODES PERTINENTS A PARTIR DE LA BASE DE TRACES/EPISODES	79
A) CALCUL DE SIMILARITE EN SE BASANT SUR LA TABLE D'INDEX DE LA BASE	79

B) CALCUL DE SIMILARITE EN SE BASANT SUR LE CALCUL DE DISTANCES ENTRE LES EPISODES EN COURS ET LES EPISODES CONCLUS	80
3. IMPLEMENTATION DU PROTOTYPE	82
3.1 PRESENTATION DES OUTILS DE DEVELOPPEMENT	82
3.1.1 MICROSOFT VISUAL STUDIO	82
3.1.2 WAMPSEVER	83
3.1.3 PHPMYADMIN	84
3.2 OBJECTIFS ATTENDUS.....	85
3.3 VALIDATION DE L'APPROCHE PROPOSEE	85
3.4 PRESENTATION DES INTERFACES DU PROTOTYPE IMPLEMENTE	85
4. CONCLUSION	88
CONCLUSION GENERALE.....	89
BIBLIOGRAPHIE.....	90

Liste des figures

Fig.1 : Définition d'un EIAH.....	05
Fig.2: Utilisation des traces dans les EIAH	07
Fig.3 : Modèle de trace.....	17
Fig.4: Architecture du Système à Base de Traces.....	17
Fig.5: Fonctionnement générale d'un Système à Base de Traces modélisées	29
Fig.6 : Le Modèle du langage UTL	30
Fig.7 : Le Modèle du langage UTL2.....	31
Fig.8 : Classification des types de traces décrivant des parcours d'apprentissage proposée par le projet TRAILS.....	33
Fig.9 : Architecture générale du Système à Base de Trace	35
Fig.10 : Représentation cognitive de la réutilisation Le Modèle du langage UTL	41
Fig.11 : Les différentes composantes du modèle de traces Musette	48

Fig.12 : Le cycle Pixed	49
Fig.13 : La visualisation de modèle de traces dans ATER	50
Fig.14 : Architecture globale du système ATER	51
Fig.15: Architecture générale du système SRTn	56
Fig.16 : Architecture fonctionnelle du système SRTn.....	57
Fig.17 : La phase de préparation	58
Fig.18 : collecte, filtrage et modélisation de traces	60
Fig.19 : Diagramme de collecte de traces.....	62
Fig.20 : Le modèle d'une trace	62
Fig.21 : Diagramme de transformation de traces	63
Fig.22 : Structure générale d'un épisode	64
Fig.23 : Différents composants de la table d'index	69
Fig.24 : La phase d'exploitation.....	69
Fig.25 : Exemple d'une trace visualisée	75
Fig.26 : Modem de traces	76
Fig.27 : Exemple de construction des épisodes	79
Fig.28 : Les épisodes en cours et les épisodes extraits à partir la base de traces	80
Fig.29 : La page de démarrage de Visual studio 2010	82
Fig.30 : Accueil WAMPSEVER	83
Fig.31 : Interface de PhpMyAdmin3.4.5.....	84
Fig. 32: Base des épisodes	85
Fig. 33: interface de calcul de similarité	86
Fig. 34: Exemple 1	86
Fig.35 : Exemple 2	87
Fig. 36 : Exemple 3	87

Liste des tableaux

Tableau.1 : Taxonomie des Traces	10
Tableau.2 : Différences entre les SBC et les SBT.....	28
Tableau.3 : Typologie de traces utilisées par le système SRTn	59

Tableau.4: Traces numériques collectées par SRTn..... 61

Tableau.5 : calcul de similarité entres deux épisodes en cours pour le même test d'évaluation81

Résumé

Le travail de recherche présenté s'inscrit dans le domaine des environnements informatiques d'apprentissage humain. Notre recherche porte plus particulièrement sur l'amélioration de l'environnement pour l'apprentissage à distance et la généralisation d'une aide individualisée pour les utilisateurs (apprenants).

L'objectif de ce travail est d'étudier les traces numériques d'activité des apprenants dans des systèmes d'apprentissage à distance. En effet, le but fondamental est de réutiliser ces traces afin d'aider les apprenants, en cherchant des situations identiques pour ces derniers. De ce fait, Un Système de Réutilisation des Traces numériques (SRTn) est proposé. Le système SRTn transforme les traces collectées et modélisées en épisodes. Le calcul de similarité entre les épisodes permet de choisir les traces (dans la base de traces) similaires aux traces de l'apprenant en cours. Enfin, le système SRTn visualise les traces pertinentes pour l'apprenant qui a demandé l'aide du système.

ملخص

يندرج هذا العمل ضمن ميدان التعليم عن بعد. بحثنا هذا يهتم خاصة بتحسين بيئة التعليم عن بعد و تعميم مساعدة فردية للمستعملين (المتعلمين).

المدفوع من هذا العمل هو دراسة الآثار الرقمية لنشاط المتعلم ضمن أنظمة التعليم عن بعد. في الحقيقة، المدفوع الرئيسي هو إعادة استعمال هذه الآثار لمساعدة المتعلمين، و ذلك عن طريق البحث عن حالات مشابهة لهؤلاء المتعلمين. من هذا المنطلق، فقد تم اقتراح نظام لإعادة استعمال الآثار الرقمية. يعمل هذا الأخير على تحويل الآثار المجموعة و المحولة إلى حلقات. تقدير التشابه بين الحلقات يسمح باختيار الآثار (الموجودة في قاعدة الآثار) المشابهة لآثار المتعلم الحالي. أخيرا، يظهر النظام الآثار الوثيقة فقط للمتعلم الذي طلب مساعدة هذا الأخير.

Abstract

The work of research presented inscribe in the domain of e-learning. Our research carries, particularly, on the amelioration of e-learning environment and the generalization of an individualizing help for the users (learners).

The objective of this work is o study numeric traces of learners in e-learning systems. Indeed, the fundamental purpose is to reuse these traces for helping the learners, searching any identical situation for these learners. By the way, a numeric traces reuse system (SRTn) is proposed. The SRTn system transforms, the collected and modelized traces, in episodes. The calculation of similarity between episodes allows choosing the traces (in traces base) similar to those of the learner in course. Finally, the SRTn system visualizes the pertinent traces to the learner asking for the help of system.

Introduction Générale

Lorsque l'on parle de l'apprentissage à distance, on est dans le domaine des Environnements Informatiques d'Apprentissage Humain (EIAH). Il n'est pas inutile de préciser le sens donné à ce mot. Un EIAH défini comme « un environnement qui intègre des agents humains (étudiant ou enseignant) et artificiels (informatique) et leur offre des conditions d'interaction, localement ou à travers les réseaux informatiques » [Balacheff, 97]. Il offre ainsi un ensemble d'outils informatiques en réseau qui permettent la gestion et la diffusion de la formation : accéder, partager ou transférer des connaissances, l'accès à des sources par téléchargement ou en consultation sur le net, et la communication entre ses acteurs, etc.

Il est organisé en fonction des besoins individuels ou collectifs (demandes individuelles, insuffisances de l'offre traditionnelle sur le territoire, problèmes de mobilité des personnes, etc.) et selon des objectifs déterminés [FOAD, 05]. Il implique l'autonomie de l'étudiant, une certaine liberté dans son rythme de progression et d'acquisition des savoirs et des savoir-faire.

Dans le cadre des EIAH, l'apprenant ne se trouve pas face à face avec son enseignant, pour lui déterminer ses questions et interrogations sur les connaissances présentées. Il n'est pas non plus en mesure de savoir si la démarche entreprise dans la construction du cours s'adapte au niveau des apprenants. C'est pourquoi, il est nécessaire de rassembler un ensemble d'informations sur l'apprenant, afin qu'on ait un bilan de l'utilisation du système d'apprentissage, pour assurer l'individualisation de la formation et son amélioration. L'objectif dans un système d'apprentissage est donc, de fournir un contenu ou un schéma adapté pour les apprenants.

Pour atteindre cet objectif, de nombreuses recherches sur l'analyse et l'interprétation des activités réalisées par les apprenants durant leurs interactions avec l'environnement de formation ont été menées. Ces travaux portent sur le recueil et l'interprétation en cours de session, d'informations appelées « *traces* ».

Les traces d'apprentissage ou les interactions avec le système d'apprentissage effectuées par les apprenants, sont de grande importance dans ce cas. En effet, ce sont les observables ou les enregistrements de l'interaction de l'apprenant dans le système d'apprentissage. Elles sont collectées, enregistrées, analysées et visualisées. Ces traces peuvent être importantes et sujettes à une réutilisation par le système.

L'aide apportée par la réutilisation des traces, ne se limite pas aux seuls utilisateurs que sont les enseignants, car les apprenants peuvent profiter encore davantage de l'aide que peut constituer la réutilisation des traces (aider les apprenants à s'aiguiller dans un contenu ou vers une activité qui peuvent leur apporter un apprentissage). En effet il existe dans ces traces d'utilisation des informations sur les chemins de tous les apprenants. En particulier, les traces d'apprentissages réussis et les traces d'échecs d'apprentissages peuvent être distinguées. En réutilisant ces traces, des chemins qui peuvent conduire les apprenants à réussir leur apprentissage seront proposés, et ceci, sans avoir à utiliser obligatoirement la modélisation du domaine faite par les enseignants.

Pour cela nous proposons un système de réutilisation des traces des apprentissages, intégré à un EIAH. Ce système consiste à proposer aux apprenants de suivre les traces similaires des apprenants qui ont réussi à atteindre leurs objectifs (réussir un test d'évaluation par exemple), il s'occupe donc à sélectionner les traces pertinentes de l'ensemble de traces collectées et représentées.

Cependant, notre objectif est donc de définir les règles inhérentes à toute réutilisation et répondre à un certain nombre de questions à savoir comment réutiliser les traces, pour qui cette réutilisation sera bénéfique et quelles traces vont être réutilisées ?

Notre mémoire est composé de cinq chapitres, arrangés de la façon suivante :

Dans le premier chapitre, nous donnons un état de l'art dont on parle sur les EIAH en générale, les définitions des concepts sur lesquelles nous nous sommes basés, pour fonder notre idée. Ensuite, nous ferons le passage vers les systèmes à base de traces, et ses caractéristiques.

Le deuxième chapitre est la suite de notre état de l'art. Nous concentrons ici, sur les systèmes qui tracent les interactions utilisateur /système, passant aux différentes utilisations des traces dans les systèmes d'apprentissages.

Dans le troisième chapitre nous parlons de la réutilisation des connaissances en générale et la réutilisation de traces particulièrement, nous achevons le chapitre par un nombre important de travaux existants concernant la réutilisation.

Le quatrième chapitre concerne la conception de notre système. Nous présentons les objectifs principaux à atteindre et les architectures générale et fonctionnelle, du système, puis nous détaillerons chaque partie de l'architecture à part.

Dans le dernier chapitre, nous parlerons de la validation et de l'implémentation des formules proposées au cours de la phase de conception des différents composants du système présenté.

Nous terminerons notre travail par une conclusion générale et quelques perspectives.

*Chapitre I : Les traces
dans les EIAH*

1. Introduction

L'apprentissage en ligne réunit les acteurs, apprenants et enseignants, à travers une plateforme numérique. Le développement rapide des Technologies de l'Information et de la Communication a modifié de façon radicale l'enseignement à distance et l'a démocratisé. Ayant connu un essor considérable avec l'Internet, cette forme d'enseignement intègre aujourd'hui des outils, dont les traces numériques, qui la rendent plus accessible, plus performante et plus proche des utilisateurs.

Les services proposés aux apprenants sont en effet devenus un critère de qualité pour les plates-formes de formation à distance, où l'utilisation des traces numériques permet d'opérer un suivi efficace et personnalisé des individus.

La plupart des systèmes de télé-enseignement fournissent cependant à leurs utilisateurs des traces de l'activité d'apprentissage. Ces traces consistent soit en une production explicitement associée à la tâche d'apprentissage (production d'un texte, réponse à une question) soit à des indices souvent implicites et de niveaux variés reflétant le comportement, les actions ou les caractéristiques de l'apprenant. Les traces seules se révèlent souvent insuffisantes au chercheur pour analyser finement les situations d'apprentissage. En effet, ces environnements n'ouvrent pas toute la palette d'informations que l'on peut obtenir lors d'un enseignement traditionnel. Il faut alors diversifier les sources d'informations : collecte indirecte par le biais d'enquêtes et de questionnaires, collecte audiovisuelle au moyen d'enregistrements visuels des situations d'apprentissage, et l'ensemble de ces informations, de ces traces sont alors utilisées dans divers contextes : régulation des situations d'apprentissage, assistance au suivi des tuteurs ou bien encore analyse des situations d'apprentissage par des chercheurs.

Dans le présent chapitre nous allons faire un tour d'horizon sur les traces et leurs exploitations dans les EIAH. Nous commençons par une définition des environnements d'apprentissage humain. Ensuite, nous présentons les différents types, exploitations et traitements sur les traces d'apprentissage. Nous passons à un ensemble de définitions des différents concepts utilisés dans ce mémoire, et nous achevons le chapitre par une conclusion.

2. Traces Dans Les EIAH

2.1. Environnement d'apprentissage humain (EIAH)

Les **Environnements Informatiques pour l'Apprentissage Humain (EIAH)** sont des environnements informatiques qui ont pour objectifs de favoriser ou susciter des apprentissages, de les accompagner et de les valider. Les réalisations sont nombreuses, mais les succès très inégaux. La recherche dans ce domaine est née avec l'informatique mais se sont surtout développées dans le sillage de l'intelligence artificielle dans les années 70. De nombreuses dénominations ont été utilisées. Le terme EIAH est né dans les années 90 [http, 1].

La recherche sur les EIAH est fondamentalement pluridisciplinaire, en appelant à la coopération de différents secteurs de l'informatique (génie logiciel, réseau, la modélisation des connaissances et des interactions, etc.), et des sciences de l'homme et de la société (psychologie, didactique, ergonomie, sciences des langages, sciences de la communication, etc.) [http, 1].

Tchounikine [Tchounikine, 02a] a donné une définition générale et simple pour l'EIAH (**Fig.1**), il le définit comme l'intention didactique et l'environnement informatique, tout en précisant l'intention didactique aux artefacts prédéfinis et les artefacts à créer.

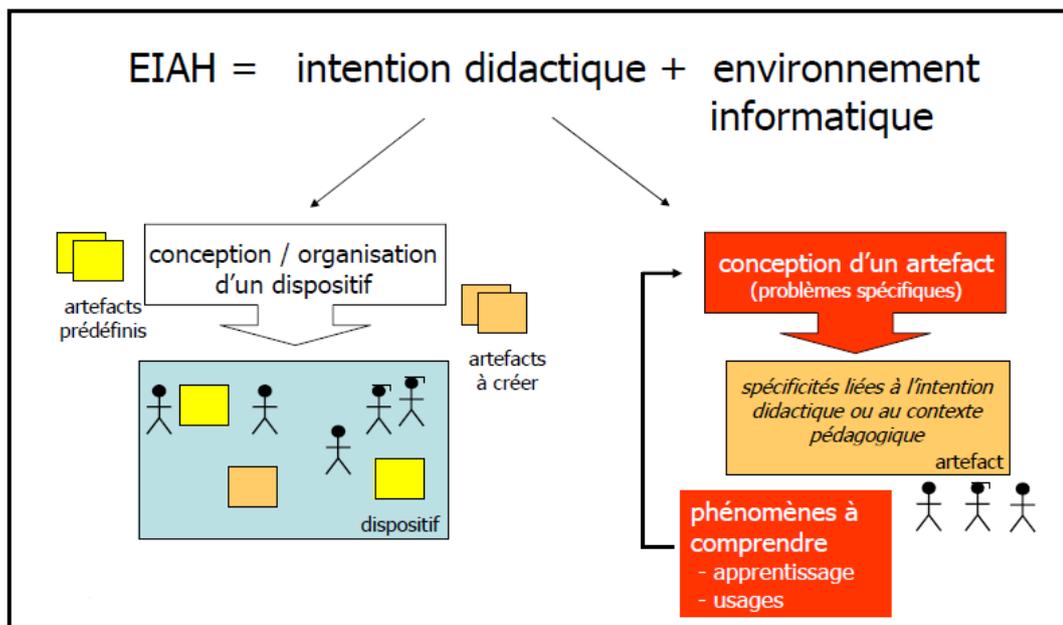


Fig.1 : Définition d'un EIAH [Tchounikine, 02a].

2.2. Exploitation des Traces dans les EIAH

- Pour quelles raisons utilise-t-on des traces ?

L'exploitation des traces d'apprentissage varie selon les environnements informatiques pour l'apprentissage humain (EIAH). A partir de traces reflétant l'activité des apprenants, des indices ont mis en évidence des comportements communs chez les apprenants [Tchounikin, 02a]. Ces indices permettront à l'enseignant-analyste de découvrir les suites d'actions qui ont tendance à mener à la réussite ou à l'échec d'un exercice [Tchounikine, 02a].

Alors l'exploitation des traces permet non seulement de comprendre le comportement de l'apprenant, ou d'un groupe d'apprenants, qui utilise un EIAH. Mais aussi, de pouvoir fournir à l'enseignant ou au formateur une information précise et adéquate pour ses besoins propres sur l'évolution individuelle et collective des apprentissages. Pour cela, la personnalisation doit prendre en compte outre les caractéristiques des utilisateurs (apprenants ou formateurs), celles des tâches ou des problèmes, ainsi que celles du contexte de leur traitement (fonctionnalités particulières et caractéristiques techniques de l'EIAH, caractéristiques sociales, institutionnelles ou économiques de la formation) [http, 3].

L'ensemble des traces peut utiliser dans divers contexte : régulation des situations d'apprentissage, assistance au suivi des tuteurs, profils d'apprenants ou bien encore analyse des situations d'apprentissage par des chercheurs, la visualisation de l'activité de l'apprenant pour le tuteur, l'élaboration d'un modèle de connaissance, la construction du modèle cognitif de l'apprenant, l'évolution de l'activité ou encore l'analyse de l'interaction entre apprenants dans des environnements d'apprentissage **collaboratifs** [Gwenegan, 05].

La Figure (**Fig.2**) montre l'utilisation d'une trace d'interaction dans les EIAH. Elle décrit la boucle générale de personnalisation des EIAH par différents acteurs en utilisant les traces. Les participants à l'activité d'apprentissage interagissent avec un EIAH, individuellement ou en groupes. Suivant leur rôle respectif, les participants ne tireront pas partie des traces de la même manière [Settouti et al, 06].

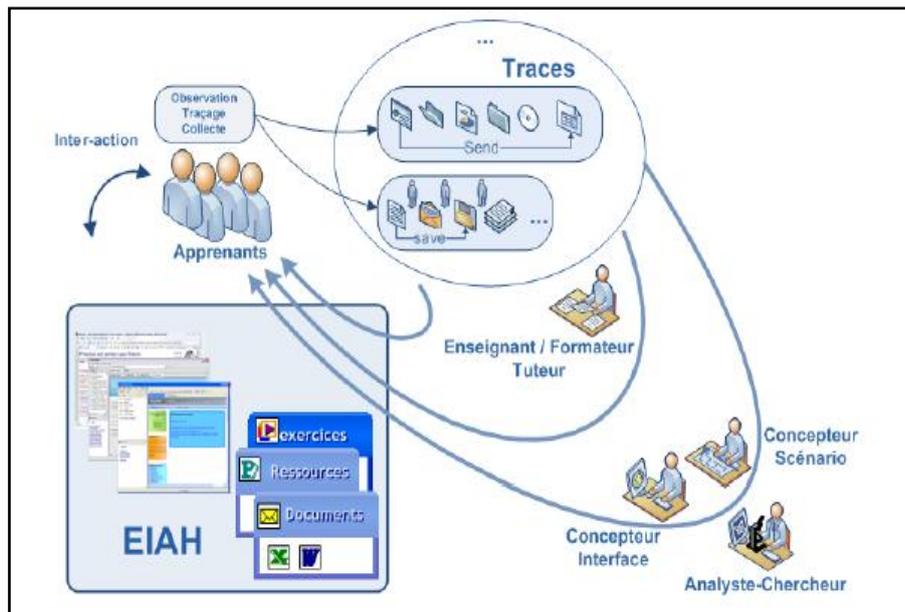


Fig.2: Utilisation des traces dans les EIAH [Settoui et al, 06].

On s'intéresse dans ce papier, aux traces numériques, considérées comme l'histoire interactionnel d'un apprenant utilisant un EIAH ; pour qu'un autre apprenant, qui a les mêmes '*caractéristiques*' que l'apprenant générant ces traces, pourra les exploiter au cours de son apprentissage afin d'atteindre son but.

2.3. Types de traces

Il existe notamment trois types de traces, traces numériques issues des artefacts informatiques, traces d'observations qui sont le résultat d'une observation directe d'une situation d'apprentissage et les traces hybrides qui sont une fusion des deux types de traces précédentes, et comme on est intéressé seulement aux traces numériques, on va se concentrer dans le reste de ce document sur ce type de traces.

L'apprentissage en ligne génère des traces, variables selon les environnements d'apprentissage, sous la forme de données personnelles à l'apprenant et de données concernant son activité [Fischer, 05] (ingénierie des EIAH):

- fichiers d'accès à la plate-forme d'enseignement
- nombre de pages de cours consultées
- nombre de messages envoyés (au tuteur, aux enseignants, à l'administration, aux autres étudiants)
- participations aux forums de discussion
- résultats des tests d'évaluation

- détails sur les choix et les parcours de formations suivies

2.4 Classification des traces numériques

Selon cette diversité de traces numériques et à partir de la lecture des articles de recherches, des rapports de mastère, des mémoires et des thèses dans ce contexte de recherche, on trouve qu'il existe plusieurs classifications pour les traces numériques.

2.4.1 Classification selon les interactions entre utilisateurs

En fonction des diverses interactions possibles, une première classification dans [Fischer, 05] est comme suit :

- apprenant-enseignant, apprenant-tuteur,
- apprenant-environnement matériel (pages Web, évaluations, exercices),
- apprenant-apprenant dans certains cas, elles peuvent être de l'apprenant vers lui-même, ou de l'apprenant vers un autre apprenant.

Ces traces numériques peuvent être stockées au sein d'une base de données, afin d'être réutilisables pour construire des analyses fines de l'attitude de l'apprenant dans le processus d'enseignement à distance.

Par exemple, l'analyse de la fréquentation par l'étudiant des cours proposés permet de déceler une absence de connexion (et donc une absence sur la plate-forme), ou de réorienter quelqu'un qui poserait une question précise.

Dans le premier cas, l'analyse peut conduire à lui envoyer un message lui demandant si cette absence de connexion est liée à un problème d'accès, ou à des questions d'ordre personnelles. Ceci a pour but de ne pas laisser une situation de difficulté s'instaurer si elle existe vraiment, ou du moins de montrer à l'apprenant qu'il n'est pas abandonné dans l'univers virtuel d'une plate-forme d'enseignement.

Dans le second cas, l'analyse du parcours de l'étudiant offre la possibilité de répondre à des problèmes ponctuels : si un étudiant pose une question et que le tuteur ou le professeur sait que la réponse se trouve dans les enseignements, la consultation des traces numériques (historique des pages consultées par exemple) offre la possibilité de réorienter l'étudiant vers un module ou un cours qu'il n'aurait pas consulté.

Les traces numériques ne concernent pas seulement les interactions provenant des apprenants : elles peuvent aussi être utiles pour garder en mémoire les réponses apportées à certaines questions par les enseignants ou les tuteurs.

L'analyse de ses interactions sert à définir des modèles et des stratégies précises d'accompagnement des apprenants dans un système d'enseignement à distance, afin de le rendre plus efficace. Ce type de modélisation des échanges sur une plateforme permet notamment de modifier les principes de fonctionnement internes : une analyse du nombre d'échanges entre un tuteur et les étudiants dont il est responsable peut ainsi conduire à redéfinir le nombre d'étudiants pris en charge par celui-ci.

2.4.2 Classification selon le contenu

Une deuxième classification, ou plus précisément une taxonomie des traces, proposée dans [Gwinigan, 05] présentée dans le **tableau.1**, est une autre proposition pour les types des traces, ils ont extrait quatre types de traces numériques : les traces informatives, les traces liées à l'exploitation d'une ressource, les traces associées à l'activité d'apprentissage et les traces attachées à l'activité de communication.

Pour réaliser un traitement de traces, il paraît nécessaire de construire une taxonomie de ces traces pour pouvoir, par la suite, définir le traitement correspondant à un type de trace similaire [Gwinigan, 05].

Dans la taxonomie proposée, les traces ont, tout d'abord, été regroupées suivant leur type (à gauche). Puis, la deuxième entrée qui a été posée c'était les classes d'exploitation : caractère de la situation (individuelle ou collective), valeur d'usage (qualitative ou quantitative) et cadre d'exploitation (étude du comportement ou étude de la connaissance), (en haut). Enfin, une dernière colonne a été ajoutée pour préciser une exploitation possible de la trace.

Une fois le cadre de la taxonomie posée, le remplissage se fait par le repris des articles qu'ils ont utilisé pour construire la taxonomie. Ainsi, pour chaque trace relevée, ils ont inscrit une marque à l'endroit correspondant.

Par exemple, l'enregistrement de la réalisation d'une action par un apprenant intervient dans le cadre d'une situation d'apprentissage individuel. De plus, la trace obtenue est

considérée comme portant une valeur qualitative dans le cadre d'une étude des connaissances de l'apprenant.

A la fin de cette opération de relecture, ils ont décidé de proposer un élément de mesure de la fréquence de l'usage de la trace, le symbole « + », plutôt qu'un résultat chiffré. Par conséquent, l'absence du « + » signifie que l'usage n'a pas été trouvé ou alors de façon non significative sur l'ensemble des articles de recherche.

De même, la présence d'un « + » montre l'existence d'un usage significatif de la trace dans le contexte défini. Quant à la marque « ++ », elle met en évidence un usage très important de la trace dans le contexte défini.

Cette proposition de taxonomie est exploitée dans le cadre de l'élaboration de l'état de l'art du projet DPULS [Gwinigan, 05].

TYPE	TRACES	Indiv	Coll	Qual	Quant	Comp	Conn	Exploitation possible
INFORMATIONS	Informations personnelles (âge, genre, ...)	++	+					Identifier l'apprenant
	Informations techniques (IP, browser, SE, ...)	++	+					Identifier l'apprenant
RESSOURCES	Nom (référence) de la ressource traité par un apprenant à un moment donné	++	+	+			+	Informers/Tenir au courant le tuteur
	Nombre d'accès	++			++		+	Dégager des informations sur l'apprenant (capacités, compétences, ...)
	Durée de consultation (/connexion)	++			++	++	+	Evaluer le niveau d'activité
	Origine de l'accès à la ressource	++			++		++	Dégager le cheminement conceptuel
	Historique du parcours des ressources	++			++		++	Dégager le cheminement conceptuel / profil d'apprenant
Activité d'APPRENTISSAGE	Qualité d'une production	++		++			++	Evaluation de l'assimilation de concepts/connaissances
	Temps de réponse à une question	++			++	+	+	Evaluation
	Tests antérieurs	++		++			++	Capitalisation pour permettre une adaptation
	Réalisations d'actions (ex : exécution, débogage)	++		+		+	+	Suivi du travail de l'apprenant / Respect du scénario
Activité de COMMUNICATION	Nombre messages envoyés (mail)		++		++	+		Améliorer la régulation d'un groupe
	Nombre messages reçus (mail)		++		++	+		Améliorer la régulation d'un groupe
	Nombre messages postés (forum)		++		++	+		Evaluer l'interaction dans un groupe
	Nombre messages lus (forum)		++		++	+		Evaluer l'interaction dans un groupe
	Fréquence des messages (sur une période donnée)		++		++	+		Améliorer la régulation d'un groupe / Evaluer la réactivité
	Destinataire des messages		++			+		Evaluer l'interaction dans un groupe
	Message de communication (contenu)		++	+		+	+	Evaluer la qualité des interactions
	Demande d'aide en ligne	++	+	+	+		+	Evaluer l'adéquation d'une ressource / Fournir une aide
	Demande d'assistance au tuteur	++	+	+	+		+	Régulation de la situation

Tableau.1 : Taxonomie des Traces [Gwinigan, 05].

❖ **Légende :**

Indiv : Individuelle ; Coll : Collective ; Qual : Qualitative ; Quant : Quantitative ; Comp : Comportement ; Conn : connaissance.

2.5. Traitement de traces numériques

Avant de parler sur les différents traitements des traces numériques, il est indispensable de connaître la forme (la représentation) de la trace à traiter.

Les Environnements Informatiques pour l'Apprentissage Humain (EIAH) génèrent différents types de traces d'interaction des utilisateurs avec leurs environnements d'apprentissage, ces traces sont généralement sous une forme non/peu exploitable directement par les utilisateurs, c'est pour cette raison qu'une représentation sous forme de fichier de log a été proposée ; mais de nombreux projets utilisent le format XML pour représenter les traces [http, 3].

2.5.1 Approches fondées sur l'analyse des fichiers logs

Le fichier log, également appelé « log », désigne le fichier présent sur le serveur qui enregistre certaines caractéristiques concernant les pages web visitées par l'internaute. Il enregistre également les incidents, les référentiels, les informations... relatifs à l'environnement de l'utilisateur, etc. L'analyse de ce fichier permet de mieux connaître les internautes visitant un site (possibilité de connaître les pages qu'il a parcourues, les mots clés qu'il a utilisés dans sa requête sur les moteurs de recherches, etc.) et ainsi de mieux répondre à leurs attentes. Ce fichier est en principe daté et classé par ordre chronologique. Un logiciel spécifique est utile pour le traitement de ce fichier. Un fichier log peut se traduire en français par « journal » [http, 5].

Le travail présenté dans [Iksal et Choquet, 05a] s'appuie sur les fichiers de log pour l'acquisition des traces, et suggèrent le métalangage Usage Tracking Language (UTL) pour analyser le comportement des utilisateurs dans l'objectif d'adapter et d'améliorer les scénarios et ressources d'apprentissage. Cette analyse s'appuie sur une approche orientée modèles décrivant les scénarios pédagogiques, et permet d'aiguiller les apprenants vers les scénarios les mieux adaptés à leur situation.

Dans le même ordre d'idée, [Stermsek et al, 07] cherche à exploiter les fichiers de *log* des apprenants et les métadonnées des pages HTML pour déduire leur profil et ainsi proposer des mécanismes de filtrage d'informations. L'extraction des données pertinentes au sein du fichier de *log* et leur analyse donnent des indicateurs sur le comportement de l'utilisateur ;

celui-ci est interprété (à partir de données statistiques, temporelles, etc.) pour dériver le profil de l'utilisateur.

Cependant, les approches existantes de gestion de traces qui sont fondées sur les fichiers de *log* doivent faire face à certaines barrières relatives à l'accès et à la signification des informations [Broisin et Vidal, 07] :

- Elles sont renfermées dans des fichiers qui ne sont généralement pas disponibles à n'importe quel utilisateur ; les administrateurs Réseau et Système sont de plus en plus méfiants et n'autorisent que très rarement l'accès à ce type d'information.
- Elles ne transcrivent pas nécessairement l'exploitation réelle du système par les utilisateurs : un grand nombre de moteurs de recherche tels que Google ou Altavista sont basés sur des robots qui analysent le contenu d'un logiciel serveur afin d'en extraire les informations pertinentes.

Même si certains systèmes différencient les visites effectuées par les robots de celles effectuées par les utilisateurs, la majorité d'entre eux accumulent le nombre de visites, ce qui ne rend pas compte de l'usage véritable du système d'apprentissage.

- L'acquisition de ces données doit être effectuée selon un intervalle de temps déterminé par le concepteur de l'outil de supervision, elle n'est pas réalisée en temps réel.
- Les informations renfermées dans ces fichiers ne sont pas toujours très précises. Les données produites par les applications s'exécutant sur le poste client (telles que les applets Java ou les fichiers Macromedia Flash qui génèrent des contenus multimédias plus riches que des applications "ordinaires") ou celles issues d'applications produisant du contenu dynamique (comme PHP ou Perl qui est basé sur la technologie CGI) ne sont pas recueillies, empêchant ainsi de connaître les informations réellement consultées par les utilisateurs.

Pour pallier aux inconvénients présentés par les approches fondées sur l'analyse des fichiers de *log*, d'autres approches proposent des mécanismes d'observation spécifiques à un outil d'apprentissage particulier.

2.5.2 Approches fondées sur des mécanismes d'observation spécifiques à un outil d'apprentissage particulier

L'application eMediathèque de la classe virtuelle eLycée [eLycée, 11] est un outil de travail collaboratif interactif qui vise à améliorer les activités réflexives lors d'un apprentissage collaboratif.

Toutes les interactions de l'utilisateur avec cet outil sont observées et traitées à partir de deux modèles [Michel et al, 05]: l'un définit les objets observables (les outils mis à disposition tels que le navigateur internet ou la messagerie instantanée, mais aussi les ressources telles que les textes, les images, etc.) et l'autre spécifie les actions réalisables par l'utilisateur (création, modification, suppression de contenus, etc.).

Les traces sont ensuite créées en fonction des activités des utilisateurs, stockées dans un composant interne de l'application, et affichées en temps réel à l'utilisateur.

Dans le cas du Knowledge Pool System (KPS) de la fondation ARIADNE [Ariadne, 96], chaque interaction d'un utilisateur avec l'outil SILO (l'interface entre les utilisateurs et le KPS) est enregistrée dans un fichier au format XML [Najjar et al, 04]. Chaque enregistrement donne des informations telles que le titre des ressources ayant été indexées, téléchargées ou supprimées, l'utilisateur à l'origine de l'enregistrement, la date de création de l'enregistrement.

Cependant, des opérations complexes telles que le dépouillement du fichier, sa décomposition et la mise en relation des différentes informations doivent être effectuées pour calculer les statistiques liées à un objet pédagogique particulier.

De plus, ce fichier est centralisé dans chacun des viviers institutionnels qui constituent le KPS global ; des efforts supplémentaires sont donc nécessaires pour obtenir une vision générale de l'utilisation des objets pédagogiques au sein du vivier global. Les traces collectées par les systèmes mentionnés ci-dessus présentent un format qui leur est spécifique, de telle sorte qu'elles ne puissent pas être réutilisées.

Pour combler les lacunes fondées par les approches précédentes, les chercheurs dans ce contexte, proposent des modèles plus généraux.

2.5.3 Approches fondées sur des modèles généraux

Dans le projet *Trèfle*, des éléments nécessaires sont représentées, par rapport à son cadre de recherche : utilisateurs, procédé et objet sous forme d'un graphe, afin de tracer l'usage d'une application, ainsi, les différents liens entre ces éléments ont été représentés.

Les éléments du modèle, ont également, une structure homogène et fortement interconnectée. Le graphe évolue au fur et à mesure de l'utilisation de l'application fournissant ainsi une base toujours plus riche de capital d'expérience que l'on peut réutiliser [Gwinigan, 05].

Les concepteurs du modèle *Trèfle* [Egyed et al, 03] ont commenté : « *En général une trace est un sous-graphe du graphe global G. Nous pouvons filtrer les traces, grâce à des graphes potentiels pour obtenir des épisodes. Un épisode est une liste ordonnée de nœuds concrets du graphe global G, construite par l'instanciation d'un graphe potentiel et organisé selon un critère d'ordre total (le temps par exemple). L'épisode contenant tous les procédés et objets concrets et la trace linéaire brute* ».

Dans Le projet SBT [Settouti et al, 06] une modélisation générale, pour la représentation de traces, a été proposée en se basant sur les modèles de temps, la trace est considérée comme une séquence temporelle d'observés (Le terme séquence temporelle dénote l'existence d'une relation d'ordre organisant les données de la trace par rapport à un repère de temps et le terme *observés* dénote que les données de la trace sont issues d'une observation) toujours accompagnée d'un modèle capitalisable [http, 3].

Le modèle de trace est un ensemble d'objets étiquetés représentant le vocabulaire de la trace [http, 3].

Cette approche permet de fusionner et manipuler des traces aux formats hétérogènes, provenant de plusieurs sources.

Dans le projet **ICALTS** (*Interaction and Collaboration Analysis' supporting Teachers and Students' Self-regulation*), et le projet **IA** (*Interaction Analysis – Supporting participants in technology based learning activities*) les traces sont considérées comme des objets pédagogiques [IA, 05].

BROISIN et VIDAL dans [Broisin et Vidal, 07] proposent une approche conduite par les modèles pour la gestion des traces d'activité des utilisateurs au sein de systèmes d'apprentissage hétérogènes instrumentés par les technologies Web, en se basant sur l'approche de modélisation CIM, qui atteint leur objectif de partage et de réutilisation de traces issues de systèmes hétérogènes car son approche par niveaux d'abstraction offre

plusieurs avantages incontestables [Choquet et Iksal, 08], et l'architecture WBEM (Web Based Enterprise Management) qui propose une organisation distribuée du cadre de supervision .

Un modèle UML générique de traces permet de structurer et d'ajouter une sémantique claire aux données observées, auquel est associée une architecture distribuée et décentralisée favorisant le partage et la réutilisation des traces collectées par différents outils et services.

3. Définitions des concepts utilisés dans ce mémoire

Ce paragraphe présente les concept-clés permettant de comprendre le reste de travail. Pour cela, nous définissons d'abord les environnements d'apprentissage humain, nous présentant ensuite la notion de traces et de traces modélisées. Nous nous intéressons par la suite aux objets et activité d'apprentissage que nous allons voir en détail leur utilité dans notre contexte au cours de chapitre de conception.

3.1. Objet d'apprentissage

Le Comité pour les Normes en Technologie d'Apprentissage (IEEE-LTSC: Learning Technologies Standard Comity) suggère cette définition : 'Learning Objects are defined here as any entity, digital or non-digital, which can be used, re-used or referenced during technology supported learning.' [http, 4] « Les objets d'apprentissage sont définis comme des entités, numériques ou non numériques, qui peuvent être utilisées, réutilisées ou référencées pour l'apprentissage assisté par ordinateur. »

Cette définition permet de considérer comme objet d'apprentissage un document imprimé, un cours, un exercice, une étude de cas, une présentation, mais également une salle de cours, un rétroprojecteur, etc.

3.2. Activité pédagogique

« Une activité est réalisée par un ou plusieurs acteurs tenant chacun un rôle à l'aide d'un ensemble de ressources (ou objets pédagogiques) accessibles au sein d'un environnement. Un acteur peut être un apprenant ou un membre de l'équipe pédagogique. » [Pernin, 03] Dans notre cas l'acteur et notamment un apprenant.

3.3. Trace

D'après P. Jermann [Jermann, 04], une trace est une observation ou un enregistrement de l'interaction de l'apprenant avec un système en vue d'une analyse.

Dans le même sens, J-P. Pernin [Pernin, 05] définit une trace comme un indice de l'activité des acteurs d'une situation d'apprentissage, qu'elle soit ou non instrumentée. Il complète, par ailleurs, sa définition en précisant qu'il s'agit d'un résultat obtenu au cours ou au terme d'une activité, d'un événement ou d'un ensemble d'évènements relatifs au déroulement de la situation d'apprentissage.

Dans une optique légèrement différente, P-A. Champin [Champin et al, 04] parle d'une séquence d'états et de transitions représentant l'activité de l'utilisateur : «la séquence temporelle des objets et opérations mobilisés par l'utilisateur lorsqu'il utilise le système est appelée trace d'utilisation ».

Dans ces trois définitions, Une trace est une trace d'activité, d'utilisation, d'interaction. On parle alors de traces premières, brutes, de base ou de bas niveau.

3.4. Trace modélisée

La naissance de traces modéliser s'apparu afin de rétrécir la diversité des ressources du web.

Une trace modélisée dans un SBT est décrite par un modèle d'utilisation et un ensemble d'instances de ce modèle, où chaque instance est située dans l'axe du temps [Djouad, 08].

Une autre définition, plus précise est mise en place, dans le même contexte, par [Laflaquière et Prié, 07] : « on appelle *trace modélisée*, ou M-trace, l'association d'une collection d'observés temporellement situés structurée par leurs relations et d'un modèle explicite de cette collection ».

Settouti dans [Settouti et al, 07], a mis la définition suivante : « Une m-trace est une trace numérique obéit à un modèle de trace, qui décrit les objets qui en font partie (**Fig.3**).

- **Modèle de trace**

On appelle *modèle de trace* le vocabulaire de la trace. Le modèle de trace permet la compréhension de la trace en en décrivant abstraitement les éléments [Settouti et al, 07].

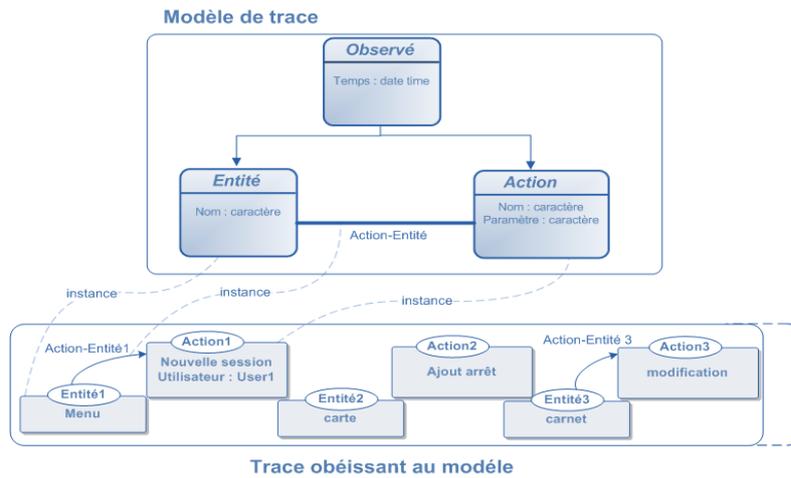


Fig.3 : Modèle de trace [Settoui et al, 07].

3.5. Système a base de trace (SBT)

Le concept de système à base de traces (SBT) est le résultat d'une thèse financée dans le cadre de la tâche « Production de traces et représentation » du projet « Personnalisation des EIAH » du cluster Rhône Alpes ISLE.

Un système à base de traces (**Fig.4**) est un système informatique permettant et facilitant l'exploitation des traces. Il décrit l'ensemble du processus de traitement de la trace, on parle ici du cycle de vie de la trace ou de théorie de la trace. Settoui dans [Settoui, 06] définit la trace comme une séquence temporelle d'observés, sachant qu'un observé constitue toute information structurée issue de l'observation d'une interaction.

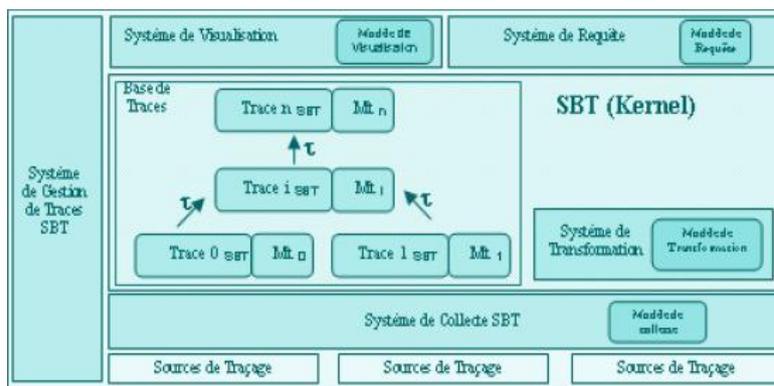


Fig.4: Architecture du Système à Base de Traces [Settoui, 06].

4. Conclusion

Nous avons pu à travers ce chapitre avoir une idée sur les différents traitements faites sur les traces d'apprentissage, les barrières existantes lors de l'accès aux informations portées dans les traces, ainsi que nous avons exploré quelques travaux existés pour achever ces barrières. Nous explorons aussi, des définitions des concepts utilisés dans le reste de notre étude.

L'utilisation des traces numériques peut apporter plusieurs avantages aux différents utilisateurs d'un environnement d'apprentissage. Avant toute utilisation des traces, un processus de collecte est indispensable. Ensuite, des processus de structuration et d'analyse des traces peuvent être utilisés afin de bien bénéficier des traces collectées.

Dans le chapitre suivant, nous allons montrer les types des systèmes traçant de l'interaction utilisateur/ système d'apprentissage, ainsi que les différents utilisations des traces d'apprentissage dans les EIAH. Nous allons mettre la lumière sur une catégorie de traces, qui est la *trace modélisée*, son intérêt, ses modèles existants et sa mémorisation, en explorant, par la suite, une des méthodes d'indexation que nous allons l'exploiter dans la suite de notre travail.

Chapitre II :
L'utilisation des traces
dans les EIAH

1. Introduction

Lorsqu'il s'agit de l'apprentissage en ligne, Il est important d'offrir à l'utilisateur humain des moyens adaptés à sa pratique, lui permettant de s'approprier l'outil informatique lors de la réalisation de ses activités. Toute activité menée à l'aide d'un logiciel conduit l'utilisateur à laisser des traces dans un fichier, dont il est parfois possible d'inférer des faits intéressants. Depuis que l'enseignement par ordinateur existe, les chercheurs ont ainsi recueilli et analysé les traces que les apprenants laissaient lorsqu'ils utilisaient les logiciels, soit pour suivre ou assister le travail effectué, soit dans le but d'améliorer les systèmes mis en œuvre ou la compréhension des phénomènes liés à leur utilisation.

Depuis quelques années, l'analyse des traces s'est développée et de nouveaux champs de recherche sont apparus concernant la nature et, surtout, la structuration des données recueillies, les modèles, méthodes et techniques pour analyser et visualiser les indicateurs construits, notamment dans le but de favoriser des apprentissages.

Ce chapitre présente les différentes utilisations de traces d'apprentissage dans les EIAH et les différents types de systèmes traçant l'activité d'apprentissage, et met en valeur la notion et le principe des systèmes à base de traces modélisées.

2. Utilisation des traces dans les EIAH

Dans le contexte des EIAH, la notion d'une trace d'activité peut signifier deux choses : (1) l'histoire interactionnelle d'un apprenant utilisant un EIAH ou (2) les productions qu'il a laissées lors de son apprentissage [Settouti, 11]. La trace peut être vue et considérée comme un élément passé restant et résultant de cette activité.

Les participants à l'activité d'apprentissage interagissent avec un EIAH, individuellement ou en groupes. Suivant leur rôle respectif, les participants ne tireront pas partie des traces de la même manière.

Par exemple, un enseignant-tuteur peut guider l'activité individuelle ou collaborative en essayant de comprendre les dysfonctionnements éventuels par rapport au scénario qu'il avait préconisé. Il peut alors adapter la session, introduire des aides personnalisées, fournir des supports pédagogiques adaptés aux différents publics.

Un enseignant-concepteur peut exploiter les traces pour personnaliser un scénario pédagogique, permettant ainsi de réguler le déroulement d'une session d'apprentissage en tenant compte de certains aspects qui ne peuvent être mesurés qu'en cours de session, tel le temps de réponse à un exercice.

L'apprenant peut visualiser sa trace et se faire une image de son évolution dans l'activité, ce qui lui permettra de comprendre son cheminement dans la construction de sa connaissance.

Ces quelques exemples d'utilisation des traces par les différents acteurs donnent une vue très parcellaire de la puissance de l'exploitation des traces car les participants peuvent avoir des besoins très différents. Il serait donc difficile de décrire exhaustivement l'ensemble des fonctionnalités imaginables.

Settoui [Settoui, 11] a établi les utilisations déjà expérimentées des traces dans des « EIAH utilisant des traces ». Cette présentation se fera en considérant quatre catégories d'acteurs, elle nous permettra de faire ressortir les éléments et besoins les plus importants pour les Systèmes à Base de Traces proposé par ce même auteur.

2.1 Traces pour l'enseignant

Les enseignants-tuteurs animant des activités pédagogiques au sein de certain EIAH sont parfois désorientés par le manque d'éléments perceptibles permettant le suivi des activités des apprenants.

Certains travaux ont mis en évidence la problématique du « manque de conscience » de l'enseignant des activités de ces apprenants pendant une situation d'apprentissage médiée.

En effet, dans une situation présentielle classique, l'enseignant a toujours des feedbacks visuels et/ou exprimés par les apprenants (questions, commentaires) qui lui permettent de réagir de différentes manières.

Pour pallier cette différence de repère, les traces sont utilisées pour reconstituer des éléments de perception de l'activité. Ces éléments sont obtenus en effectuant un certain nombre de manipulations sur les traces (extraction, filtrage, calculs et visualisation).

Il est important de noter que d'un tuteur à un autre les éléments de perception peuvent être très différents, certains étant par exemple préoccupés par les éléments liés à la performance des apprenants, d'autres à la communication qui s'instaure dans l'activité d'apprentissage collaborative.

a- Le calcul d'indicateurs

Une trace présentée de manière très synthétique sous forme de bilan peut permettre de fournir à l'enseignant une analyse objective du travail d'un apprenant. Utilisés pendant ou après l'activité d'apprentissage, des indicateurs simples peuvent décrire des informations précises et ponctuelles, comme par exemple le nombre d'accès à une ressource en ligne, le nombre d'essais pour chaque réponse, le temps passé dans chaque activité, le taux de réussite.

Beaucoup d'EIAH permettent la visualisation de statistiques calculées à partir des traces. Par exemple l'environnement **ESSAIM** [Després, 01] permettant au tuteur de voir à base de traces d'interactions, le degré d'interaction de l'apprenant ainsi que le temps qu'il a passé dans chaque étape.

L'application **Combien?** [Le Calvez et al, 03] présente des statistiques pour chaque exercice fait (durée pour trouver la solution, nombre d'erreurs ou d'exercices achevés).

Synergo [http, 2] dans un contexte d'usage collaboratif permet lui aussi de visualiser entre autres le nombre de messages émis et le nombre d'objets manipulés.

Le système **DIAS** [Bratitsis et Dimitracopoulou, 05] (Discussion Interaction Analysis System) permet la visualisation d'indicateurs (degré d'interaction, taux de contribution) pour mieux soutenir la collaboration.

Le système **Virtuoso** [Stefanov et Stefanova, 05] envoie automatiquement un courriel au tuteur contenant les statistiques relatant entre autres les scores obtenus et les erreurs commises ainsi que les différentes traces utilisées après une session d'apprentissage.

Certes, les renseignements tirés de la trace sont importants pour l'enseignant qui veut s'assurer par exemple que ses étudiants ont parcouru le matériel pédagogique, mais ils ne révèlent rien sur la nature de ce parcours. En fait, l'utilisation des traces comme une source de représentation d'information quantitative révèle par exemple que l'apprenant a fait quelque chose, mais ne montre pas comment il s'y est pris [Settoui, 07].

Dans la plupart des cas, il est en effet nécessaire pour un enseignant-tuteur voulant intervenir sur une situation de connaître le cheminement effectué par l'apprenant pour accomplir sa tâche. Ceci permettra par exemple de détecter, pendant l'activité, que

l'apprenant s'est égaré et éloigné de sa tâche ou bien *a posteriori* d'expliquer à l'apprenant la cause de son succès ou de son échec [Settoui, 07].

b- Les parcours des apprenants

Le parcours de l'apprenant montre d'abord sa progression au sein de la séquence de tâches proposée lors d'une séance d'apprentissage (ressources pédagogique à lire, exercices, etc.).

Le niveau de détail des actions disponibles dans la trace permettant de suivre la progression d'un apprenant peut être discuté et en soi un problème complexe.

Une trace brute montrant la succession des actions élémentaires opérées sur l'interface (Click_bouton_Ok) n'est pas pertinente. Celle-ci serait en effet inexploitable par le tuteur en situation de suivi car les éléments produits par les apprenants au niveau de l'interface sont trop éloignés de la sémantique de leur activité d'apprenants.

Dans ce sens, [Labat, 02] est intéressé aux EIAH fournissant des informations d'un plus haut niveau d'abstraction aidant le tuteur à apprécier le travail de manière quantitative et surtout qualitative.

Même si la majorité des EIAH tracent les parcours des apprenants (en inscrivant dans une trace généralement les actions et les objets manipulés), peu d'entre eux permettent une visualisation cohérente de ce parcours synthétisée au bon niveau d'abstraction.

Par exemple, le projet **FORMID** [Guéraud, 04] offre à l'enseignant une visualisation de l'état d'avancement des apprenants en utilisant un modèle de tâche permettant le traitement et la structuration de la trace au bon niveau de besoin du tuteur.

CourseVis [Mazza et Dimitrova, 03] et **VLE** [Hardy et al, 04] permettent de visualiser le parcours d'un apprenant pour cerner son comportement.

CollabLogger [Morse et Steves, 00] permet de voir les parcours d'un groupe d'apprenants collaborant.

D'autres EIAH donnent la possibilité à l'enseignant-tuteur de rejouer les traces d'interaction et de reprendre le parcours d'un apprenant pour comprendre son comportement ou les problèmes survenus lors de sa session d'apprentissage, par exemple **DREW** (Dialogical Reasoning Educational Web tool) [Baker et al, 03], Combien? [Le

Calvez et al, 03] SimPLE (Simulated Process in a Learning Environment) [Plaisant et al, 99] ou APLUSIX CoIAT [Avouris et al, 07] permet quant à lui de visualiser les traces de parcours non pas directement dans l'EIAH mais dans une interface dédiée, en synchronisation avec d'autres sources d'observation multimédia (vidéo).

2.2 Traces pour l'apprenant

Bon nombre d'EIAH tracent les interactions entre apprenants et environnement, sous la forme de fichiers souvent non exploitables directement par cet apprenant.

Il existe deux grandes utilisations de la trace au service de l'apprenant : (1) la visualisation de sa propre trace et (2) l'assistance à base de traces.

La première est concernée par la présentation des traces dans un format qui fait sens pour l'apprenant dans le cadre de son activité d'apprentissage (approche *Mirroring*).

La seconde concerne le traitement automatique des traces par l'EIAH pour guider l'apprenant dans son activité (approche *Guiding*).

a. Trace comme miroir de l'activité de l'apprenant

Les traces d'interaction d'un apprenant donnent une image fidèle de son activité. Elles permettent de refléter ce qu'il fait avec un EIAH lui permettant ainsi de prendre conscience de la tâche qu'il effectue [Settoui, 07]. L'apprenant peut revenir sur son activité et peut revoir les questions et les hypothèses qu'il a formulées tout au long de sa session d'apprentissage [Settoui, 07].

Même si la majorité des EIAH tracent l'apprenant et permettent la visualisation de cette trace, ils ne lui permettent pas toujours d'interagir avec. Quelques systèmes permettent cependant cette interactivité.

Par exemple, **SimPLE** (Simulated Process in a Learning Environment) [Plaisant et al, 99] propose aux apprenants une visualisation interactive de leurs actions permettant d'annoter certaines parties et d'en éditer des éléments.

APLUSIX [Nicaud, 87] permet aussi à l'apprenant d'interagir avec sa trace en parcourant les calculs qu'il a effectués. L'apprenant peut ainsi corriger lui-même les exercices qu'il n'a pas résolus correctement en reprenant la résolution et agir donc sur sa trace.

b. Trace comme facilitateur de l'activité de l'apprenant

Les traces peuvent être utilisées pour assister l'apprenant dans son activité d'apprentissage. Cette approche a été proposée dans le cadre du système PIXED [Heraud, 02]. Ce système propose aux apprenants consultant un cours en ligne de réutiliser le

parcours d'apprentissage d'autres apprenants en se fondant sur les principes du Raisonnement à Partir de Cas [Mille, 06]. PIXED permet de renvoyer à l'apprenant une possibilité d'exploiter des traces d'apprentissages comme source de connaissances pour assister son orientation dans la progression d'apprentissage dans un cours. La trace est considérée comme un cas composé d'une séquence d'actions qu'on peut réutiliser.

Dans le cas d'utilisation des traces à des fins d'assistance comme MUSETTE [Champin et al, 04] l'EIAH utilise les traces mais sans les présenter directement à l'apprenant ou l'enseignant. Des calculs faits sur traces récupérées permettent de modifier les interfaces. En s'aidant du modèle de l'utilisateur ou du profil d'apprenant extrait en partie de la trace d'utilisation, un système conseillé peut guider l'utilisateur dans son activité. Un système conseillé peut même être instancié par des calculs sur les traces pour obtenir l'état de l'utilisateur comme le montre [Settouti, 07].

2.3 Traces pour l'enseignant-concepteur d'activités pédagogiques

Une des nouvelles évolutions des EIAH offre la possibilité de scénariser l'activité pédagogique de l'apprenant. Les scénarios permettent à l'enseignant de définir et concevoir des parcours destinés aux apprenants, plus ou moins complexes et adaptés à ses objectifs pédagogiques. L'approche par les scénarios permet de bâtir des situations d'apprentissage personnalisées par les tuteurs.

Si dans les systèmes existants, l'utilisation des traces se réduit le plus souvent à fournir aux apprenants et/ou au tuteur une information utile pour réguler l'activité, il est relativement rare de l'utiliser pour fournir un feedback aux concepteurs du scénario et de l'activité pédagogique.

En effet, le scénario est l'expression de la tâche prescrite par l'enseignant alors que la trace d'exécution du scénario est l'expression de la tâche effective lors de l'utilisation par les acteurs. Une analyse comparant la tâche prescrite par le scénario et la tâche effective lors de l'utilisation par les apprenants participe à la personnalisation du processus d'apprentissage en étant base d'évolution du scénario pédagogique. Les traces d'utilisation sont alors un élément d'amélioration continue de la qualité d'un scénario pédagogique.

De ce point de vue, la démarche présente certains avantages. Elle permet tout d'abord de décrire des observables à partir de la structure du scénario construit par l'équipe des concepteurs. Elle propose ensuite une façon de retranscrire ces observables dans un

langage interprétable en conception, dans le but de pouvoir comparer le scénario conçu *a priori* à celui résultant des usages observés *a posteriori*.

2.4 Traces pour le chercheur

Une des utilisations courante des traces est l'analyse par des chercheurs étudiant une situation d'apprentissage. En effet, les EIAH peuvent être utilisés par un apprenant individuellement et en complète autonomie, et/ou en classe mais médiée par un artefact informatique, à distance et/ou en collaboration avec d'autres apprenants. Cette transformation du contexte d'apprentissage amène les chercheurs à s'interroger sur la nature même de l'apprentissage.

Les traces fournissent une matière première permettant d'étudier les caractéristiques du processus d'apprentissage, les interactions et aidant à modéliser l'apprenant. De nombreux outils d'analyse existent permettant notamment l'annotation de traces, des transcriptions de vidéos et des visualisations sophistiqués comme ColAT [Avouris et al, 07], Observer XT [http, 6], MacShapa [Sanderson, 94], TRANSANA [http, 11].

3. Types des systèmes qui tracent les interactions utilisateur-système

Dans le domaine de l'interaction homme-machine, le traçage des interactions utilisateur système et l'utilisation des traces comme outils de recherche existent depuis longtemps.

Ces traces sont des historiques, utilisés pour comprendre la situation d'interaction ou pour assister l'utilisateur dans sa tâche [Ollagnier, 07].

M.Ollagnier. Beldame [Ollagnier, 07] a proposé de classifier les situations d'utilisations des histoires interactionnelles selon l'usage qui en est fait, et en particulier selon le fait qu'elles sont ou non présentées à l'utilisateur. Nous présentons dans cette section les différentes situations d'utilisation des traces informatiques classées en fonction des possibilités et des types d'opérations que le système informatique traçant permet d'effectuer sur ces traces.

3.1 Systèmes utilisant l'histoire interactionnelle sans la présente aux utilisateurs

Le premier groupe de systèmes identifié est celui des systèmes qui ne présentent pas l'histoire interactionnelle aux utilisateurs bien qu'ils l'utilisent. Ces systèmes utilisent les traces des interactions entre utilisateurs et système, mais ne les exploitent pas sous forme de visualisation. Des calculs sont réalisés sur ces traces (fichiers logs), en vue de prévoir,

conformément à un modèle implicite, les actions futures des utilisateurs et ainsi modifier l'interface pour qu'elle corresponde au comportement prédit.

Les informations relatives aux interactions sur lesquelles les calculs sont effectués correspondent aux informations de types suivants : accès à des ressources, consultations des écrans, clics effectués, temps passés aux opérations, choix effectués, réponses données aux éventuelles questions, etc. Ces traitements sont automatiques et prévus par le programme. Les actions effectives de l'utilisateur sont comparées à un modèle d'actions prévues.

De tels systèmes s'intéressent ainsi aux préférences des utilisateurs pour personnaliser l'interface. Certains de ces systèmes proposent des interfaces qui « donnent et prennent conseils » en interaction avec l'utilisateur. Ces interfaces reflètent les calculs faits sur les interactions utilisateur-système stockées en mémoire, en proposant à l'utilisateur des suggestions d'actions.

3.2. Systèmes présentant une visualisation de l'histoire interactionnelle destinée à l'analyse de la situation

Le deuxième groupe de systèmes est celui des systèmes proposant une visualisation des traces d'interactions utilisateurs-système à un analyste de la situation, qui n'est pas l'utilisateur du système.

Dans le cadre d'analyses des usages en situations interactionnelles, il peut être intéressant que l'analyste de la situation, par exemple le chercheur, ait accès aux traces des interactions entre utilisateurs et système.

Depuis longtemps, les traces informatiques sont utilisées par les chercheurs pour espionner la manière dont les sujets se comportent dans une situation donnée ou utilisent un système, qui peut précisément être le système sur lequel est installé le logiciel étudié.

Ce genre d'études a existé en ergonomie, en sciences de l'éducation, en psychologie, et en communication.

3.3. Systèmes présentant une visualisation de l'histoire interactionnelle pour l'utilisateur et lui proposant la possibilité d'y naviguer

Le troisième groupe de systèmes offre une visualisation des traces d'interactions aux utilisateurs, et leur permet de naviguer dans ces informations. Ces systèmes présentent

l'histoire interactionnelle aux utilisateurs en vue de supporter leur activité. Les possibilités des utilisateurs d'interagir avec cet historique sont limitées à de la navigation et ne concernent pas le déclenchement de nouvelles actions ni la saisie d'informations déclenchant des actions. Certains systèmes concernent la navigation, d'autres sont destinés à des situations d'apprentissage.

3.4. Systèmes présentant une visualisation de l'histoire interactionnelle destinée à l'utilisateur et lui proposant la possibilité d'agir dessus

Le quatrième groupe de systèmes que nous identifions concerne les systèmes qui présentent une visualisation de l'histoire interactionnelle à l'utilisateur et qui lui offrent la possibilité d'agir dessus. Ces systèmes utilisent l'histoire interactionnelle comme un outil pour les utilisateurs, pour entrer des données ou des commandes.

4. Systèmes à base de traces modélisées

4.1. Notion de systèmes à base de traces modélisées

Selon Settouti [Settouti, 11], La notion de Systèmes à Base de Traces modélisées (SBT) désigne une classe particulière de SBC (Systèmes à Base de connaissances). Les SBT partagent avec ces Systèmes d'une part la nécessité de modéliser les connaissances dont est constituée la trace moyennant des représentations de l'interaction de l'utilisateur exprimées à l'aide de primitives relatives à l'environnement informatique tracé. Et d'autre part, la nécessité de manipuler ces traces pour inférer de nouvelles connaissances, i.e., de nouvelles traces. Si les ambitions inférentielles d'un SBC consistent à raisonner automatiquement sur une représentation formelle, volontaire et suffisante du monde, les SBT s'inscrivent plutôt dans le cadre d'une inférence basée sur des traces d'interactions, inscrites automatiquement par ce système, mais évoluant dynamiquement durant l'activité. En effet, dans les SBT, les traces représentent l'ensemble de connaissances porteuses de sens, i.e., ayant une sémantique interprétable par la machine avec la particularité que ces connaissances s'inscrivent dans des traces pouvant être infinies, puisqu'elles sont induites par une utilisation et interaction continue avec le système considérée.

4.2 Différences entre les SBC et les SBT

SBT	SBC
- Les observées sont reçus au fil du temps d'une manière semblable aux flux de données.	- Les faits sont disponibles à la fois et généralement stockées dans une base de faits.
- Le flux des observées dans les traces n'est pas bornée dans le futur, et est potentiellement infini	- Les bases de faits sont limitées.
- Les relations entre les observées tels que l'ordre temporel ou la causalité jouent un rôle important pour l'interrogation et/ou transformation des traces	- Les relations entre les faits font généralement partie des faits (e.g. les références _a des clés étrangères dans une base de données déductive)

Tableau.2 : Différences entre les SBC et les SBT.

Les divergences (dans le **Tableau.2**) montrent bien les limites actuelles des SBC pour la manipulation de traces et rendent les langages d'interrogation et les moteurs d'inférence traditionnels non adaptées à la tâche d'interrogation et de raisonnement sur les traces. Elles engendrent ainsi le besoin de disposer d'un langage et moteur d'inférence spécifique aux traces.

4.3. Fonctionnement de systèmes a base de traces modélisées

Dans son architecture classique, un SBC comprend : (i) une base de connaissances relatives à un domaine d'application, (ii) une base de faits contenant des faits ou données caractérisant le problème courant, (iii) un programme souvent appelée moteur d'inférence, qui manipule les bases, recherche les connaissances adéquates et mène un raisonnement (suite d'inférences) pour résoudre le problème courant.

Comme étant une classe particulière des SBC, les SBT fonctionnent selon le schéma décrit par la figure (**Fig.5**) :

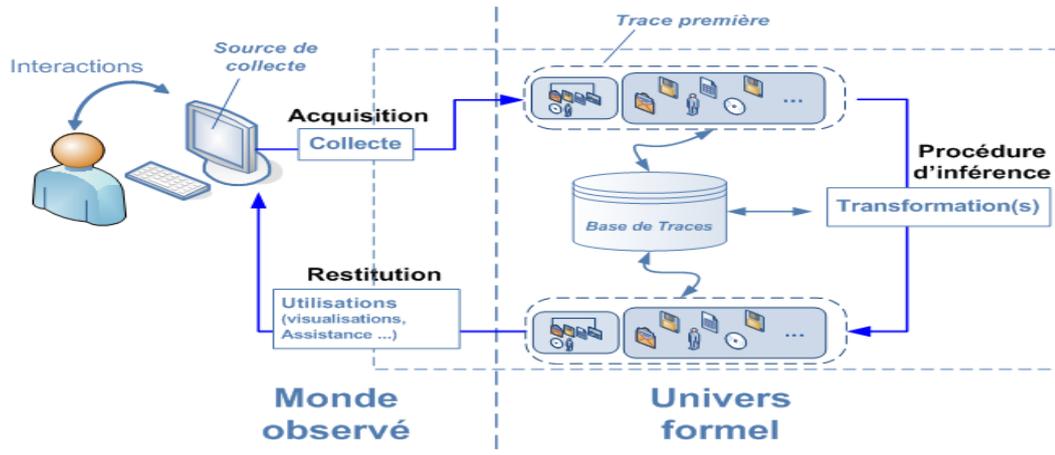


Fig.5 : Fonctionnement générale d'un Système à Base de Traces modélisées

[Settoui, 11].

- l'acquisition des traces consiste à collecter une trace qualifiée de première décrivant les interactions de l'utilisateur et le système observé. Une ou plusieurs sources de collecte (e.g., capteur logiciel, keylogger, etc.) sont utilisées pour obtenir cette trace.
- la trace première peut être transformée pour produire, par application de règles de transformation, de nouvelles traces.
- les traces transformées peuvent faire l'objet de restitutions dans l'univers observé en permettant des visualisations pouvant être interprétées par les utilisateurs concernés (l'utilisateur tracé, l'utilisateur observant l'activité, etc.)

Concevoir un SBT consiste ainsi à modéliser puis représenter des traces issues de l'observation d'une activité à l'aide d'un formalisme de représentation puis à transformer les traces par l'intermédiaire de règles de transformations pour répondre à un besoin spécifique. Le SBT permettra ainsi de fournir des services d'inférence à l'aide des règles de transformations raisonnant sur un ensemble de traces. Ces services d'inférence se baseront sur des représentations de modèles de traces, et des langages pour spécifier les requêtes et les transformations sur les traces.

5. Modèles et représentations considérées pour les traces

5.1. Le modèle du langage UTL

Un des projets mettant explicitement au centre de la réingénierie des EIAH la problématique de l'exploitation des traces est le projet ReDim (Reengineering Driven By Models). Dans ce cadre, Choquet et ses collègues estiment que les traces peuvent être considérées comme des objets pédagogiques, au même titre que les scénarios ou les ressources pédagogiques. Le projet de recherche REDiM considère l'enseignant comme le principal acteur de la conception et de la réingénierie d'un EIAH mais confrère le traitement des traces d'utilisation à trois rôles différents : le concepteur (l'enseignant), l'analyste et le développeur. L'enseignant, qui est le mieux qualifié d'un point de vue pédagogique pour spécifier ce qu'il faut observer dans une session d'apprentissage, doit disposer des outils pour appréhender, dans son univers métier, les résultats de cette observation. Pour assister l'enseignant dans sa démarche de réingénierie, Choquet et ses collègues ont défini un métalangage de modélisation et de construction de traces baptisé UTL (Usage Tracking Language).

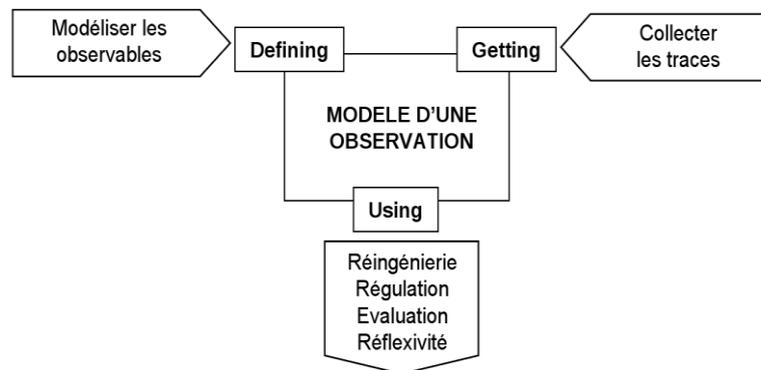


Fig.6: Le Modèle du langage UTL [Settoui, 07].

L'idée générale derrière UTL est que chaque trace d'utilisation prise en considération lors de la modélisation de l'observation doit concourir à témoigner d'un usage observé, et que ces usages observés sont à mettre en relation avec le scénario pédagogique prédictif afin de pouvoir les analyser [Settoui, 11]. UTL décrit sous trois facettes les besoins (**Fig.6**), la manière de collecter et d'utiliser les traces : la première facette <Defining> permet d'assister notamment le concepteur/pédagogue dans la phase de définition des observables dont il aura besoin, la deuxième facette <Getting> permet la description des moyens d'acquisition et de collecte des traces et enfin la troisième facette <Using> permet de définir l'utilisation

des traces (réingénierie, régulation de l'activité). De ce point de vue, la démarche présente certains avantages. Elle permet tout d'abord de décrire des observables à partir de la structure du scénario construit par l'équipe des concepteurs. Elle propose ensuite une façon de retranscrire ces observables dans un langage interprétable en conception, dans le but de pouvoir comparer le scénario conçu à priori à celui résultant des usages observés à posteriori.

Le métalangage UTL, dans sa première version, ne considère que des traces collectées de manière automatique. Il est centré sur l'importation des traces dans un format indépendant du dispositif d'apprentissage [Settoui, 11].

5.2. Le modèle du langage UTL2

Pour instrumenter la modélisation et l'analyse de l'observation dans une perspective de capitalisation, Choquet et ses collègues ont défini une nouvelle version du métalangage UTL permettant la description des méthodes d'analyse de l'observation en vue de leur capitalisation en étendant UTL avec un ensemble d'éléments dédiés à cette description. Cette nouvelle version permet par contre une description structurée des indicateurs dans une forme indépendante des formats de traces générées par un dispositif d'apprentissage et du langage de modélisation pédagogique employé pour décrire le scénario pédagogique [Settoui, 11].

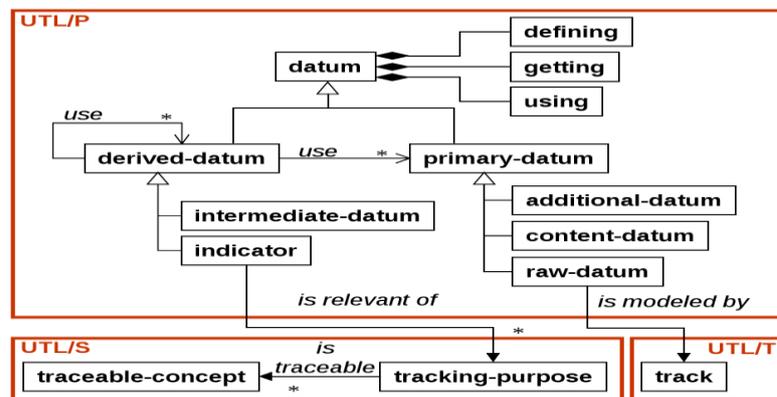


Fig.7 : Le Modèle du langage UTL2 [Settoui, 11].

Dans UTL-2 (**Fig.7**), différents types de données peuvent être modélisées dans la trace qui identifie deux types principaux de données impliqués dans l'analyse de l'observation : la donnée dérivée (derived datum) et la donnée primaire (primary datum). Les données primaires ne sont pas calculées ou établies avec l'aide d'autres données et peuvent être issues d'un fichier log ou d'une base de données. Une donnée dérivée est une sorte de production de l'apprenant. Les données additionnelles (additional-datum) sont des données

qui ne sont pas collectées à partir de l'activité de l'apprenant mais qui pourraient être utiles pour l'analyse. Une donnée dérivée est calculée à partir des autres données et peut être soit une donnée intermédiaire (utilisée temporairement pour calculer quelque chose d'autre) ou un indicateur : une information définie comme ayant une signification spéciale dans un scénario pédagogique spécifique ou situation d'apprentissage [Settouti, 11].

5.3. Classification des types de traces décrivant des parcours d'apprentissage proposée par le projet TRAILS

Un autre exemple de projet considérant les traces pour la réingénierie des EIAH est le projet TRAILS (Personalized and Collaborative Trails of Digital and Non-Digital Learning Objects). Ce projet s'est centré sur la caractérisation des traces d'utilisation dans un environnement de formation à distance sur Internet. Les membres de ce projet utilisent le terme de <parcours> (<trail>) plutôt que celui de <trace> puisque leurs travaux ne considèrent que les traces des parcours. Deux types de parcours sont identifiés : les parcours qui sont générés lors de la planification d'une activité d'apprentissage (par exemple, un scénario pédagogique) et les parcours qui émergent pendant une session (i.e., la trace d'utilisation). Le projet fait le constat que les traces de type parcours dépendent de plusieurs paramètres [Settouti, 11] :

- la nature de la situation pédagogique ;
- les possibilités d'interaction et du contexte d'apprentissage ;
- les objectifs d'utilisation des parcours (évaluation des apprentissages, régulation de la session d'apprentissage, réflexivité sur l'activité, réingénierie).

Partant de ce constat, le projet TRAILS a classé les types de traces dans une taxonomie décrites par la figure (**Fig.8**).

Les types en blanc caractérisent des parcours existants avant l'activité d'apprentissage (typiquement les traces générées automatiquement à partir des scénarios pédagogiques préconisés). Les types en noir ne s'appliquent qu'aux parcours construits pendant l'activité d'apprentissage (typiquement les traces observées de l'exécution des scénarios définis), et les concepts en gris caractérisent ces deux types de parcours. Ceci montre que le projet TRAILS considérait sur une même échelle les traces des parcours élaborés dans l'optique d'organiser la session d'apprentissage et les traces des parcours construites par l'utilisation et l'interaction avec le dispositif d'apprentissage [Settouti, 11].

Dans le contexte de la scénarisation pédagogique, cette classification des types de traces permet de reconstruire les scénarios descriptifs de l'activité menée pendant une session d'apprentissage. En plus de la réingénierie du scénario pédagogique, le projet TRAILS revendique aussi que l'apprentissage à base de traces de type parcours peut se décliner sur plusieurs points de vue (régulation par le tutorat, adaptation automatique, apprentissage réflexif, etc.) [Settoui, 11].



Fig.8 : Classification des types de traces décrivant des parcours d'apprentissage proposée par le projet TRAILS [Settoui, 11].

5.4. Le projet LISTEN

Un autre projet LISTEN (Literacy Innovation that Speech Technology ENables) [Settoui, 11]. donne de bonnes indications pour la réingénierie des EIAH par les traces d'apprentissage. Ce projet, au delà de la qualité de ses résultats, est un excellent exemple par son déroulement et l'évolution de la collecte et l'analyse des traces. Dès ses débuts, ce projet s'est structuré autour du développement et la réingénierie d'un Tutoriel Intelligent nommé < Reading Tutor > destiné à l'apprentissage de la lecture avec le principe suivant : le système écoute l'apprenant lire à haute voix et permet d'interagir avec lui afin d'améliorer sa lecture, sa compréhension moyennant des algorithmes de reconnaissance vocale [Settoui, 11].

L'usage des traces dans ce projet ne se limitait pas seulement à l'évaluation des connaissances de l'apprenant, exploite de manière indirecte, pour la réingénierie mais servait également à analyser statistiquement de vastes ensembles de données collectées pendant plusieurs sessions d'apprentissage [Settoui, 11]. Pour la représentation des traces,

les chercheurs du projet LISTEN ont créé des bases de données relationnelles spécifiques pour stocker les données collectées pendant les sessions. Ils ont également stocké les connaissances et ressources manipulées par le tuteur dans d'autres bases de données, liées avec les bases de collectes, de manière à faciliter l'analyse et la compréhension des traces. En ce qui concerne la modélisation des traces, les chercheurs de LISTEN ont identifié plusieurs dimensions pour définir le schéma des données. Puisque le dispositif d'apprentissage et les objectifs évoluent avec le temps (e.g. application d'une nouvelle technique de fouille), les chercheurs ont donc défini différents schémas de base de données en fonction de la version du système d'apprentissage (Reading Tutor) et des objectifs d'analyses [Settoui, 11]. En outre, certaines des tables de la base de données contiennent des millions d'enregistrements [Settoui, 11]. Les chercheurs ont alors privilégié la redondance des données à la normalisation du modèle conceptuel de la base, de manière à réduire le temps d'exécution des requêtes qui augmente considérablement si la requête est portée sur plusieurs tables.

5.5. L'approche système à base de traces modélisées (SBT)

La plupart des travaux précédents emploient une certaine notion de trace, et la trace a un rôle et une utilisation particulière dans chaque approche. De manière abstraite, Settoui a défini *la trace comme une séquence temporelle d'observés*. Le terme *séquence temporelle* dénote l'existence d'une relation d'ordre organisant les données de la trace par rapport à un repère de temps et le terme *observés* dénote que les données de la trace sont issues d'une observation. Cette définition permet de dire que la trace informatique est un document numérique qui révèle des données temporellement situées résultant d'une activité d'observation.

Le système à base de traces s'inspire des travaux au laboratoire LIRIS sur l'approche MUNETTE. Dans cette approche, la trace d'utilisation d'un système informatique est structurée en état-transition et est exploitée afin d'être réutilisée selon le principe du Raisonnement à Partir de Cas. Les systèmes à base de trace découlent d'une approche plus générale que MUNETTE permettant de représenter un cadre pour l'exploitation des traces. Un système à base de traces (SBT) est un système informatique permettant et facilitant la manipulation des traces. La figure (**Fig.9**) montre l'architecture générale d'un tel système dans lequel on a défini les différents composants.

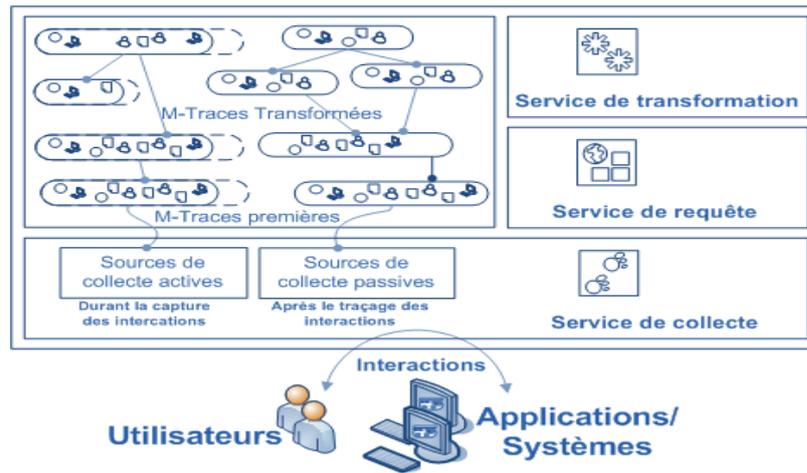


Fig.9 : Architecture générale du Système à Base de Trace [Settoui, 11].

Les traces *SBT* dans un SBT sont considérés comme une séquence temporelle d'observés et elles sont toujours accompagnées de leurs modèles de traces. Les traces les plus utilisées sont les fichiers Log mais de nombreux projets utilisent le format XML pour représenter les traces. La vidéo est aussi considérée comme une trace même si elle est utilisée la plupart du temps comme complément à d'autres traces étudiées.

- **Le modèle de trace**

Le *modèle de trace* est un ensemble d'objets étiquetés représentant le vocabulaire de la trace. Ce modèle doit être explicite et permet d'expliquer la trace. Les modèles de trace font référence à d'autres modèles de traces.

- **Le SERVICE de requête (SBT Kernel)**

Le service de requête (*SBT Kernel*) est le noyau du système. Il est constitué d'un système de transformation et d'une base de traces. La base de trace est l'ensemble des traces du SBT et de leurs modèles. Elle permet le stockage permanent et l'accès à tout moment aux traces. La base de traces peut être par exemple une collection de fichiers XML ou une base de données temporelle. L'exploitation de la trace consiste en partie en sa transformation. Le système de transformation s'occupe des différentes transformations τ requises lors de la modification et manipulation de la trace. Ce système permet de modifier la trace en enrichissant ou filtrant ses données, de modifier le modèle de trace, de mettre à jour la base de traces ou bien de faire des transformations automatiques en utilisant un modèle de transformation. Le modèle de transformation est un ensemble de règles formelles explicitées dans un certain langage.

- **Le système de collecte**

Le système de collecte est un ensemble structuré de processus permettant de convertir les données en traces en utilisant les outils adéquats. Le système de collecte permet de sélectionner des sources de traçage. En effet, la trace collectée est souvent obtenue après plusieurs retours aux sources de traçage afin de l'améliorer.

- **Le système de visualisation**

Le système de visualisation permet de visualiser les traces et facilite donc leur analyse et interprétation. Le système de visualisation doit permettre de jouer certaines traces en accédant aux sources de traçage et aux données relatives à ces traces. Ce système doit gérer et accéder aux différentes applications associées aux sources de traçage comme les vidéos. Il donne l'accès direct depuis la trace à la source de traçage pour interroger celle-ci plus directement. Par exemple, exploiter la vidéo comme trace temporalisée liée par le temps avec une trace du SBT. Le système de visualisation peut être intégré dans un système de transformation à un niveau quelconque. Le système de visualisation doit être capable de visualiser les traces issues de transformations.

- **Le système de gestion des traces**

Enfin, le système de gestion des traces concerne la gestion des modèles (modèles de trace, de transformation, de requête). Il permet l'ajout, la suppression des traces. Il regroupe les processus de conservation des traces et des droits d'administration de ces traces. Il permet de gérer le graphe de transformation des traces permettant de relier l'évolution des traces et leur cycle de vie. Il peut être considéré comme un système assurant la traçabilité des transformations effectuées sur les traces.

- **Avantage des SBT**

Contrairement aux approches existantes, l'approche SBT ne fixe pas aucun format de trace. En outre, il définit les services pouvant être effectués sur les traces dans des modalités d'exploitations in situ et à posteriori de l'activité tracée (i.e., des traitements ponctuels et continus).

Les systèmes et les utilisateurs voulant exploiter les traces disposent de langages pour modéliser, interroger et transformer des traces moyennant des règles de transformation. Ce cadre utilise des règles de transformation comme un mécanisme d'abstraction et de raisonnement, avec la même motivation et les mêmes avantages de l'usage des règles dans les systèmes de base de connaissances traditionnels. Il intègre des langages qui permettent

de spécifier des modèles, des requêtes pour l'extraction des éléments de la trace (sous forme de substitutions de variables), des transformations pour construire de nouveaux éléments dans des traces transformées, et ce, de façon ponctuelle ou continue.

6. Conclusion

L'exploitation des traces est au cœur de plusieurs pratiques guidées par différents acteurs ayant divers rôles dans l'EIAH. Il nous a permis aussi de mettre la lumière sur les différentes utilisations qui peuvent être faites de l'observation dans un cadre pédagogique. On se rend bien compte que le domaine est riche, les possibilités sont multiples et les objectifs sont divers et que les outils pour exploiter les traces doivent soutenir ces utilisations.

Dans ce chapitre, nous avons vu les différentes utilisations des traces dans les EIAH, nous avons concentré sur la notion des traces modélisées, leurs exploitations par différents EIAH. Et enfin, nous explorons quelques travaux utilisant la notion de traces modélisées.

*Chapitre III : La
réutilisation des
connaissances*

1. Introduction

La réutilisation se définit comme une approche de développement de systèmes selon laquelle il est possible de construire un système à partir de composants existants [Mechid, 08]. Cette approche s'oppose aux approches usuelles de développement dans lesquelles la construction d'un nouveau système part de zéro ("from scratch"), et nécessite de tout réinventer à chaque fois.

La réutilisation des connaissances contribue énormément dans le domaine de e-learning. Bon nombre de travaux centralisent également sur la réutilisation de « L'histoire interactionnelle » utilisé pour comprendre la situation d'interaction ou pour assister l'utilisateur dans sa tâche.

Pour bien comprendre la réutilisation en e-learning, nous avons besoin de définir le principe de la réutilisation ainsi que les connaissances supportant la réutilisation et les connaissances effectivement réutilisés.

Dans cette partie, nous nous intéressons à décrire le principe de la réutilisation de connaissances en générale et en e-learning précisément. Nous dénommons : *connaissance d'apprentissage* toute connaissance utilisée dans et/ou par un environnement d'apprentissage humain (EIAH).

Nous présentons donc la réutilisation des connaissances, puis nous décrivons la réutilisation dans les différentes disciplines de e-learning. Tout en concentrant sur la réutilisation des traces d'apprentissage, ensuite nous manifestations des études ont été menées, aboutissant à la définition de normes, des méthodes et des outils innovant dans ce domaine et qui ont été adoptés et utilisés.

2. La réutilisation

2.1 Origines et concepts

La *réutilisation*, terme ancien qui a consisté à ne pas réinventer la roue [Mechid, 08]. Au fur et à mesure qu'on élabore des solutions aux problèmes courants, ces solutions sont essayées pour des problèmes similaires. Quand une solution prouvée est utilisée plusieurs fois pour résoudre le même type de problème, elle devient alors acceptée, généralisée et normalisée.

La réutilisation fut évoquée en 1968 par M. McIlroy dans une conférence de l'OTAN sur l'ingénierie du logiciel (article re-edite dans [Mcilory 76]), suite à un échec patent de la part de développeurs pour livrer des logiciels de qualité à temps et à prix compétitifs.

La réutilisation permet de réduire les coûts et les délais de conception, d'implémentation et même de l'analyse. La valeur ajoutée d'une telle démarche est d'éviter de tout redévelopper et d'utiliser des composants existants. En effet, adapter une solution existante est souvent considéré comme plus facile et plus rapide que de concevoir à partir de rien.

Mais il faut bien signaler que la réutilisation est valable et intéressante si le coût de recherche, d'adaptation et d'assemblage des composant est nettement inférieur au développement à partir de rien (développement « *from scratch* »).

La réutilisation a été mise en œuvre d'abord au moyen des bibliothèques de code composées d'ensembles de fonctions, procédures ou blocs de code permettant d'assurer un traitement ou un ensemble de traitements cohérent.

2.2. L'apport de la réutilisation en différentes disciplines

La réutilisation a fait déjà l'objet de recherches dans différentes disciplines commençant par le raisonnement à partir de cas (RAPC) qui repose sur l'hypothèse que « des problèmes similaires ont des solutions similaires » tel que un *cas* représente un problème particulier avec sa solution. Un nouveau problème est résolu en le comparant à tous les problèmes contenus dans une *base de cas*.

Le domaine du génie logiciel constitue aujourd'hui aussi un thème de recherche à part entière ou la réutilisation de composants logiciels est d'un intérêt majeur pour le développement de logiciel. Elle présente une meilleure fiabilité des programmes développés. Ces recherches ont été élargies aux systèmes d'information et l'ingénierie du document [Maurice, 98] [Semmak, 98]. La complexité croissante des systèmes d'information et leur évolution de plus en plus rapide ont motivé un intérêt accru pour les modèles et méthodes de réutilisation.

La réutilisation a été opté aussi dans le domaine de l'ingénierie des méthodes qui se préoccupe de la définition de nouvelles méthodes d'ingénierie des SI en combinant différentes (parties de) méthodes pour développer une solution optimale au regard du problème donné.

Les méthodes existantes sont décomposées en fragments réutilisables qui ont déjà fait leurs preuves pour définir d'autres méthodes, On parle souvent de *fragment de méthode* ou encore modules ou blocs.

Les méthodes ainsi obtenues sont elles-mêmes modulaires et peuvent être modifiées et étendues facilement [Mechid, 08].

Nous citons parmi les travaux dans ce domaine:

« Une méthodologie d'aide à la conception de systèmes d'information fondé sur la réutilisation (A Reuse based method to help information systems Design) » de *Maurice Demourieux Michel*: Cette recherche aborde la réutilisation selon deux dimensions : la conception de composants réutilisables et la conception des systèmes d'information par réutilisation de composants [Maurice, 98].

Ainsi, parmi les travaux dans le domaine de l'ingénierie du document nous citons :

« Une methodologie de réutilisation en ingénierie des documents Le système SABRA (System of Authoring by Reuse based on Annotations)» de Azeddine Chikh [Chikh, 04].

Dans ce travail une méthodologie de réutilisation a été proposée, baptisée ARBRE (Approche d'aide à la Rédaction Basée sur la RéUtilisation), en plus, deux modèles ont été proposés concernant la représentation et l'organisation des composants réutilisables. Le premier, baptisé ASARD (**Annotations Structurées d'Aide à la Réutilisation de Documents**), décrivant les connaissances de réutilisation d'une part et la spécification des nouveaux besoins rédactionnels d'autre part. Le second baptisé MCD (Modèle de Composant de Document) et englobant le premier modèle, prend en charge la représentation en UML, XML et OEM des composants de documents et leurs constituants.

Parmi les travaux d'apprentissage par situation-problème (Problem Based Learning ou PBL), nous trouvons « *Conception des scénarios d'apprentissage par réutilisation des patrons* » de Samira MECHID [Mechid, 08], qui consiste à réutiliser des patrons de conception dans le développement des scénarios d'apprentissage dans les PBL représentés avec la spécification IMS LD.

Nous avons pu, à travers cette première partie du chapitre, avoir une idée sur l'intérêt de la réutilisation et son application dans les différentes disciplines.

3. La réutilisation des connaissances dans les EIAH

La réutilisation des connaissances peut être définie comme la récupération de connaissances épisodiques en mémoire et leur utilisation via un raisonnement analogique pour résoudre de nouveaux problèmes. L'identification des indices et des processus qui permettent à un opérateur de récupérer des connaissances utiles et utilisables pourrait constituer une base utile pour la conception de systèmes capitalisant l'expertise. L'analyse des mécanismes de résolution de problèmes par analogie pourrait permettre d'offrir une aide adaptée à l'opérateur (aide à la résolution de problèmes) [Chikh, 04].

Toute utilisation de connaissances pourrait être considérée comme une réutilisation : les connaissances sont en effet des structures exploitables grâce à leur élaboration et utilisation dans le passé, même si ce n'est pas nécessairement sous la même forme. Nous réservons «réutilisation» pour l'utilisation (par adaptation) de connaissances spécifiques élaborées dans le passé [Chikh, 04].

On ne considère, donc, pas être dans le cadre de la «réutilisation», quand, pour résoudre un problème spécifique, on procède à l'instanciation d'un schéma ou qu'on fait appel à une règle générale qui associe, à un problème abstrait, sa solution abstraite [Chikh, 04].

3.1. Représentation de la réutilisation

Une représentation de la réutilisation en psychologie cognitive, provenant du raisonnement par analogie [Chikh, 04], est bien illustrée par la figure (Fig.10). Le problème pour lequel on fait appel à la réutilisation, est appelé problème cible et la solution qu'on veut construire est appelée solution cible. La solution qu'on recherche à réutiliser est la solution source. Cette dernière est associée à un problème source décrivant son contexte.

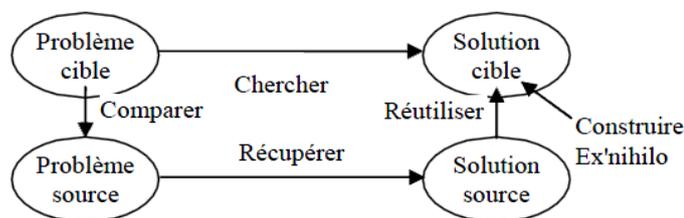


Fig.10 : Représentation cognitive de la réutilisation [Chikh, 04].

Des recherches dans [Chikh, 04], ont constaté que des sous problèmes d'un projet global peuvent être résolus, ou bien à partir de zéro, ou bien par réutilisation, selon le mode qui

est le plus intéressant d'un point de vue cognitif. Plusieurs études empiriques montrent que le principal critère de sélection entre ces deux modes est le coût de l'adaptation. Ce dernier peut être évalué sur la base du degré de ressemblance entre la solution cible et la solution source éventuelle. La confrontation des spécifications, décrites dans le problème source, avec les besoins, décrits dans le problème cible, permet de trouver la solution source adéquate, qui sera réutilisée dans la construction de la solution cible.

3.2. Réutiliser les traces d'apprentissage

Pour réutiliser une trace, il faut qu'elle soit représentée sous une forme adéquate pour être exploitée (réutilisée) par l'utilisateur (un système, un être humain, assistant ...) ainsi, elle doit être mémorisée, d'une façon permanente, en autorisant son extraction à chaque fois que l'on a besoin de la réutiliser par l'utilisateur.

Pour bien cerner cette étude, il était nécessaire de répondre aux questions suivantes : pourquoi réutilise-t-on les traces ? Pour qui cette réutilisation est ciblée ? Avant de réutiliser une trace, il faut qu'elle soit mémorisée d'abord, comment faut-il la mémoriser ? Comment les réutiliser (ou, quelles sont les outils, techniques, systèmes, ... à utiliser pour réaliser cette réutilisation) ?

a) Pourquoi réutilise-t-on les traces d'apprentissage ?

La réutilisation des traces se fait : afin de simplifier le processus d'apprentissage pour les acteurs de l'EIAH (apprenants, enseignant,...), afin d'exploiter au mieux les expériences d'autres utilisateurs et pour améliorer la tâche d'apprentissage. Par exemple, on réutilise les traces afin de permettre à un apprenant d'exploiter les traces des autres apprenants qui sont dans le même contexte d'apprentissage, cela pour faciliter cette tâche au mieux possible pour l'apprenant.

b) Pour qui la réutilisation des traces d'apprentissage est ciblée ?

Lorsque la tâche de réutilisation des traces est ciblée aux tuteurs, cela les aide dans le suivi des apprenants pour mieux les guider (exemple pratique **EM-AGIIR** : Environnement Multi-Agent de supervIsion à base d'Indicateurs Réutilisés).

Si elle est ciblée aux apprenants, elle les poussera à poursuivre au mieux leur apprentissage en suivant les expériences passées d'eux-mêmes (ce qu'on appelle la **réflexivité** ; exemple pratique **e-lycée**) ou d'autres apprenants dans le même contexte d'apprentissage (exemple pratique **Pixed**).

La réutilisation peut être ciblée aussi à un assistant logiciel (un agent généralement) avec pour objectif de réutiliser cette expérience dans ce contexte pour assister l'utilisateur à effectuer sa tâche comme l'approche de modélisation dénommée **MUSETTE** (Modéliser les USages Et les Tâches pour Tracer l'Expérience).

Nous allons présenter, par la suite, quelques uns des exemples précédents en détail.

c) Comment mémoriser les traces d'apprentissages ?

Pour réutiliser les traces, elles doivent être mémorisées de façon permanente sur un support qui permet l'accès, à tout moment, qu'on ait besoin de ces traces.

L'enregistrement des traces à réutiliser peut être centralisé (au sein des systèmes d'apprentissage), ou décentralisé (distribué), c.-à-d. distincts des systèmes à observer afin de favoriser le partage des traces, tout en offrant la possibilité de facilité d'exploitation de ces systèmes de stockage pour bénéficier de l'expérience de nombreux utilisateurs [Broisin et Vidal, 07].

4. Les travaux existants concernant la réutilisation

La complexité de conception des systèmes et leur évolution, ont motivé un intérêt accru pour une approche de développement plus efficace : la réutilisation. Cette approche permet de construire un système nouveau à partir de composants existants, éprouvés et réutilisables.

Pour montrer l'intérêt et l'importance de la réutilisation des *connaissances d'apprentissages* dans les EIAH, nous exposerons des travaux, où la réutilisation participe par excellence.

4.1. Projets de réutilisation dans le domaine des ressources pédagogiques

Nous commençons par les recherches qui se focalisent surtout sur la réutilisation des ressources pédagogiques.

4.1.1. LE PROJET CDE

CDE, pour *Course Designer Environment*, de l'école Polytechnique Fédérale de Lausanne, s'inscrit dans le grand projet de l'EPFL nommé *ClassRoom2000* dont le but est de développer des environnements auteurs pour la production de cours interactifs avec un haut niveau de partage et de réutilisation d'items didactiques.

CDE propose aussi un outil nommé *CoDes Authoring Tool* pour *Course Designer Authoring*

Tool permettant aux enseignants de créer, structurer et visualiser des cours interactifs [Rekik, 99].

4.1.2. Le système OLA

Le système OLA, pour *Online Learning Application*, était, un système développé par la société Oracle. Ce système a permis de définir la norme RCOS, pour *Reusable Content Object Strategy*, qui a été adoptée par plusieurs distributeurs de logiciel auteur, tels que la société Macromedia, pour caractériser les documents produits.

La norme RCOS définit quatre niveaux hiérarchiques de document [Hathaway et al, 97].

- Le niveau le plus bas, nommé niveau composant, est constitué de documents monomédia (texte, graphiques, audio, vidéo, fichiers exécutables).
- Le niveau des items didactiques simples, nommés RCO, qui sont formés d'une succession d'éléments du niveau composant.
- Le niveau Sujet, est constitué de documents pédagogiques simples (agrégation de RCOS) tels que des cours (premier niveau d'assemblage).
- Le niveau Groupe de Sujet, est constitué de documents pédagogiques complexes (agrégation de documents pédagogiques simples) pour former par exemple des modules. (second niveau d'assemblage).

4.1.3. Le projet ARIADNE

ARIADNE¹, pour *Alliance of Remote Instructional Authoring and Distribution Networks for Europe*, est un projet européen, regroupant principalement la Belgique, l'Espagne, la Finlande, la France, la Grande-Bretagne et la Suisse, et dont l'objectif est de définir des méthodologies et de produire des outils d'échange de données pédagogiques numériques pour les différentes catégories d'enseignement (classique, continu, à distance, etc.).

4.1.4. Le projet SEMUSDI

Ce projet de l'INSA de Rouen [Delestre et al 99, Delestre et al 98] a débuté en 1996. Le serveur

Semusdi² est conçu pour collecter, gérer, stocker, rechercher, consulter, diffuser et organiser des items didactiques pour des utilisateurs repartis sur plusieurs sites géographiques distants.

Les auteurs de SEMUSDI utilisent plutôt le terme de brique élémentaire. Ces briques élémentaires peuvent être de différentes formes : texte et hypertexte (HTML), images fixes ou animées, son, logiciel d'animation et de simulation, exercices d'application courts, exemples de cas concrets. L'ensemble de ces briques est géré au sein d'une base de données accessible via le Web.

4.1.5. SCORM

Scorm est un modèle de référence pour le partage de contenus, une spécification créée par l'ADL (US) permettant de créer des objets pédagogiques. Permet aux systèmes d'apprentissage en ligne de trouver, importer, partager, réutiliser, et exporter les contenus d'apprentissage, fondé sur le concept « d'objet de contenu partageable » puisque le contenu est séparé des contraintes liées au contexte et aux spécificités du logiciel d'exécution.

Le SCORM de ADL (Advanced Distributed Learning) s'impose aujourd'hui comme le standard en matière de conception de cours et de plates-formes e-learning (LMS).

Nous trouvons plusieurs d'autres projets toujours dans le contexte de la réutilisation des ressources pédagogiques citons : le projet **MEMORAE** de [Benayache, 05] proposant de gérer les ressources au moyen d'une "mémoire organisationnelle de formation" basée sur des ontologies, les travaux d'Issam REBAI [Rebai et Labat, 04] définissant quatre types de composants logiciels spécifiés aux EIAH (Environnements Interactif pour l'Apprentissage Humain), à savoir : Les composants logiciels pédagogiques (CLP), les composants logiciels de services (CLS), les composants logiciels techniques (CLT) et les composants logiciels de fabrication (CLF).

4.1.6. Discussion

Nous remarquons que tous ces travaux se ressemblent en définissant d'abord une unité minimale (SCO, RCO, BI, notion, brique,..) et elle est indépendante du contexte pédagogique. Le principe de la décomposition est présent dans ces projets, ainsi pour une meilleure réutilisation, ces éléments sont caractérisés par une description.

Plusieurs projets se référant à des ontologies pour une description sémantique comme le projet Memorae [Benayache, 05].

La plupart des travaux se sont focalisés sur la réutilisation des ressources et donc des composants techniques. Nous pensons ainsi que le besoin d'améliorer la réutilisation des ressources pédagogiques est à présent relativement satisfait.

4.2. Projets sur réutilisation des traces d'apprentissage

Nous explorons, dans le reste de ce chapitre, d'autres recherches qui se centralisent effectivement, sur la réutilisation des traces d'apprentissages ou en d'autres termes les historiques ou encore les expériences d'apprentissage.

M. Beldame dans [Ollagnier, 07] distingue deux catégories de systèmes réutilisant des traces: les systèmes présentant l'histoire interactionnelle à l'utilisateur avec des possibilités de simple navigation, et les systèmes présentant l'histoire interactionnelle avec des possibilités d'interactions sur les entités constitutives de la trace et sur la trace elle-même. Les systèmes appartenant à chacune de ces deux catégories mettent en place une visualisation.

4.2.1. LifeLine

Dans la première catégorie, on trouve la représentation *LifeLine* de [Plaisant et al, 96] qui permet facilement au lecteur de remettre en contexte des éléments ponctuels d'une histoire ou de corréler certains évènements grâce à des lignes de vie superposées sur un axe temporel linéaire horizontal.

4.2.2. L'EIAH Simple

D'autres formats de l'expérience d'utilisation peuvent être mis à disposition de l'utilisateur.

Dans l'EIAH *Simple*, les apprenants échangent des segments vidéo de leurs actions permettant de partager plus facilement leur expérience [Plaisant *et al*, 99].

4.2.3. PadPrints

D'autres formats de l'expérience d'utilisation peuvent être mis à disposition de l'utilisateur.

Dans l'EIAH *Simple*, les apprenants échangent des segments vidéo de leurs actions permettant de partager plus facilement leur expérience [Plaisant et al, 99].

4.2.4. LE système FootPrints

Avec *PadPrints*, une carte de navigation représente cette histoire. Parfois, c'est l'expérience d'utilisation de plusieurs utilisateurs qui est visualisée [Hightower et al, 98].

4.2.5. L'environnement DREW

Les études faites avec l'environnement DREW (*Dialogical Reasoning Educational Web tool*) [Séjourné et al, 04] montrent que les apprenants exploitent les traces d'activité (*chat* de DREW en particulier) au profit des processus réflexifs de l'apprentissage. C'est en situation collaborative que les élèves sont capables de revenir sur leurs traces, et de co-construire un diagramme qui organise et hiérarchise les arguments élaborés pendant l'activité pédagogique. Il est intéressant dans le cas de cette étude de voir à quel point les auteurs de la trace deviennent alors acteurs d'un certain nombre de transformations de celle-ci : sélection, suppression, réduction, construction.

Nous trouvons plusieurs autres projets toujours dans le contexte de la réutilisation des traces d'apprentissage citons :

4.2.6. MUSETTE

Le modèle MUSETTE (*Modélisation des Usages et des Tâches pour Tracer l'Expérience*) est un modèle générique de traces. Il permet de décrire l'utilisation qui est faite d'un système informatique [Laflquière et al, 05]. Ce modèle peut être appliqué au domaine des EIAH. Il se compose du *modèle d'utilisation* qui comprend l'ensemble des *objets* observables du système et des *opérations* que l'on peut effectuer sur ces objets [Laflquière et al, 05].

Lorsqu'un utilisateur manipule ces objets avec des opérations, il laisse ce qu'on appelle une *trace d'utilisation*. Sur la figure (**Fig.11**), la trace montre que l'utilisateur était sur la « Page1 » et avait à sa disposition 2 liens. En cliquant sur « Lien1 », il a fait une transition vers la « Page2 ». Les traces d'utilisation peuvent être comprises grâce à des *modèles de tâches*. Un *modèle de tâche* est « une restriction du modèle d'utilisation décrivant les propriétés de ses objets qui sont toujours vérifiées lors de la réalisation de la tâche en question. Il peut aussi être accompagné d'explications sur le rôle des éléments du modèle d'utilisation impliqués dans cette tâche ». Lorsqu'on arrive à comprendre une partie de la trace d'utilisation grâce à un modèle de tâche, cette partie est appelée *cas d'utilisation*. Par l'extraction de *cas d'utilisation*, MUSETTE permet de comprendre l'utilisation qui est faite du système observé. Pendant l'exécution d'une tâche, le raisonnement à base de cas permet de fournir une aide contextualisée à l'utilisateur sur l'utilisation du système [Champin et al, 04].

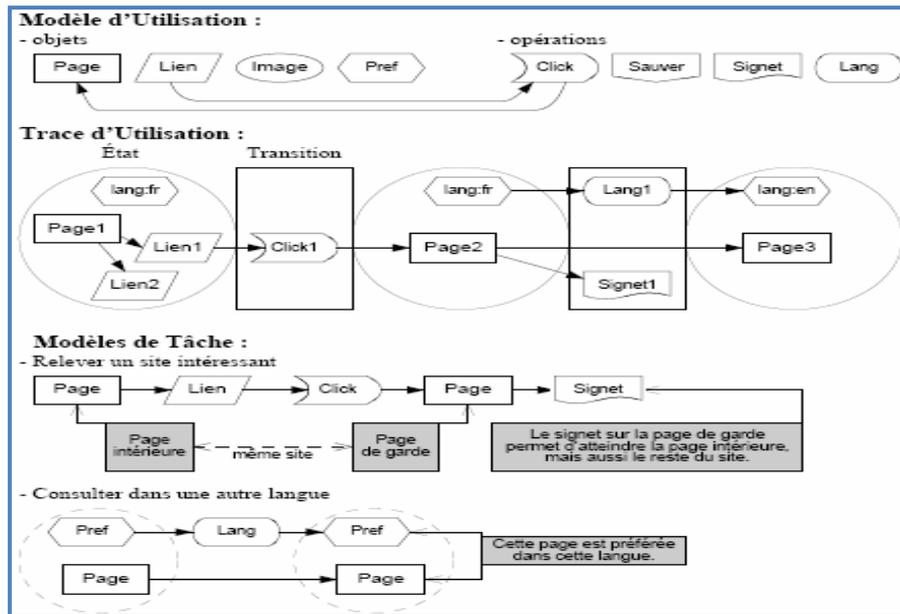


Fig.11: les composantes du modèle de traces Musette [Champin et al, 04].

4.2.7. Le projet EPICéA

Le projet EPICéA (Évaluation de Processus Interactifs de Capitalisation d'Épisodes d'Apprentissage) [Ollagnier, 03] a pour objectif de concevoir et d'évaluer sur les plans cognitif et ergonomique un environnement d'apprentissage en ligne intégrant un module de raisonnement à partir de cas (RÀPC par la suite). Le but de ce projet est d'associer des connaissances théoriques en intelligence artificielle, concernant notamment le RÀPC, à l'expérience de S3R dans le domaine de la formation en ligne et à distance. L'idée est de capitaliser l'expérience d'apprentissage des apprenants pour enrichir le système au fil des différentes utilisations qui en sont faites, pour l'apprenant lui-même (à partir de ses utilisations précédentes du système) ou pour d'autres apprenants en situations d'apprentissage similaires auxquels le système fait profiter de l'expérience des premiers apprenants. Ce travail de recherche comporte trois objectifs majeurs [Ollagnier, 03]:

- Déterminer comment réutiliser les expériences d'apprentissage dans un système de formation en ligne et à distance.
- Étudier l'impact sur l'apprentissage de l'intégration du modèle de RÀPC dans le système conçu afin d'étudier la pertinence de l'utilisation de ce paradigme en contexte d'apprentissage.
- Apprécier la valeur sur le plan de l'ergonomie cognitive de la représentation de ce paradigme dans le système, et notamment à l'interface homme-machine.

Dans le projet EPICéA, les expériences sont capitalisées d'apprentissage des apprenants en les intégrant au système e-cursus à l'aide de l'outil RÀPC, afin d'améliorer et d'enrichir le système au fur et à mesure des différentes utilisations qui en sont faites. L'idée est de tracer les actions des apprenants dans leurs utilisations de la plate-forme. Ces traces seront découpées en épisodes d'utilisation du système, c'est à-dire des cas d'apprentissage ou « grains d'expérience ». Leur contenu éducatif dépendra de la nature des objets d'intérêt pédagogique choisis. Ces épisodes intégreront des informations sur le contexte d'apprentissage, liées à l'apprenant, au système et à la situation d'interaction.

4.2.8. Le projet PIXED

Dans le cas de l'EIAH *PIXED* [Heraud, 04], l'apprenant peut voir l'état de son parcours pédagogique grâce à un *réseau notionnel annoté* permettant de construire un "chemin d'expérience" pour le guider dans ses choix de ressources pédagogiques. Ce chemin d'expérience est une représentation transformée de la trace d'apprentissage selon un objectif d'orientation de l'activité d'apprentissage. Cette représentation explicite et structurée du passé de l'activité favorise l'auto évaluation, l'auto interrogation sur certains concepts précédemment abordés, et la révision si nécessaire.

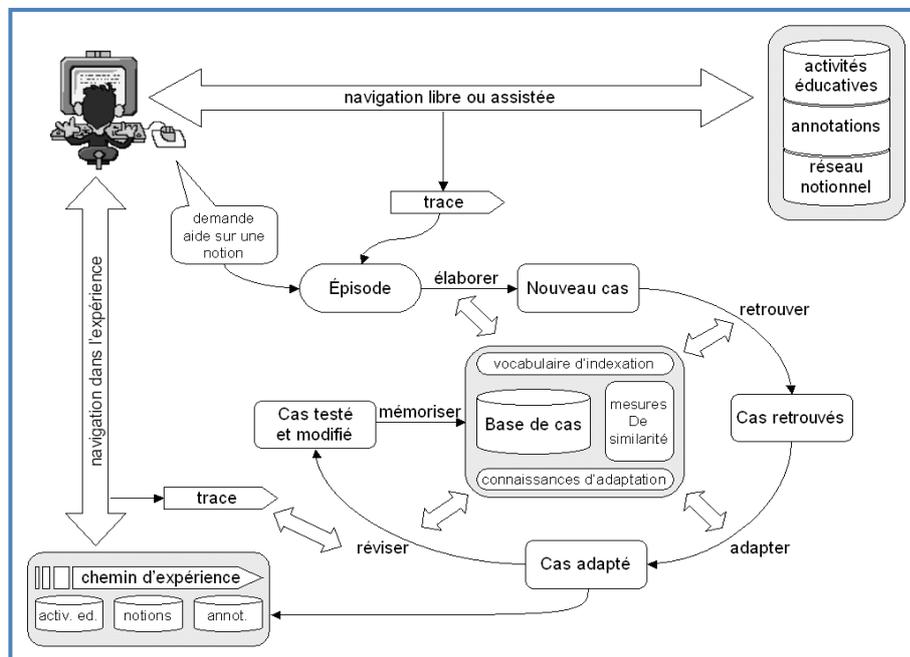


Fig.12 : Le cycle Pixed [Heraud, 04].

Dans Pixed (Fig.12), la navigation dans l'expérience n'étant pas l'unique mode de navigation, l'apprenant peut utiliser le système dès la mise à disposition d'activités

éducatives. Pixed va tracer et mémoriser les épisodes de navigation comme présenté dans la figure. Il amorce ainsi la base de cas. Les attributs descriptifs qui caractérisent l'épisode constituent le vocabulaire d'indexation. Ces attributs sont [Heraud, 04]:

- La notion « objectif d'apprentissage », sa maîtrise initiale et sa maîtrise objectif.
- Les notions consultées et les maîtrises de ces notions.
- Les activités éducatives jouées.
- Les couples notion-QCM joués et leurs résultats.
- Les annotations.

4.2.9. Le système ATER

ATER est une application implémentant la notion de systèmes à base de traces modélisées (SBT). ATER permet de collecter, transformer, requêter, visualiser et de partager des traces modélisées en tant que représentations de connaissances. ATER supporte l'expression de modèles de trace comme ontologies OWL décrivant des concepts et relations structurant les éléments de la trace (**Fig.13**).

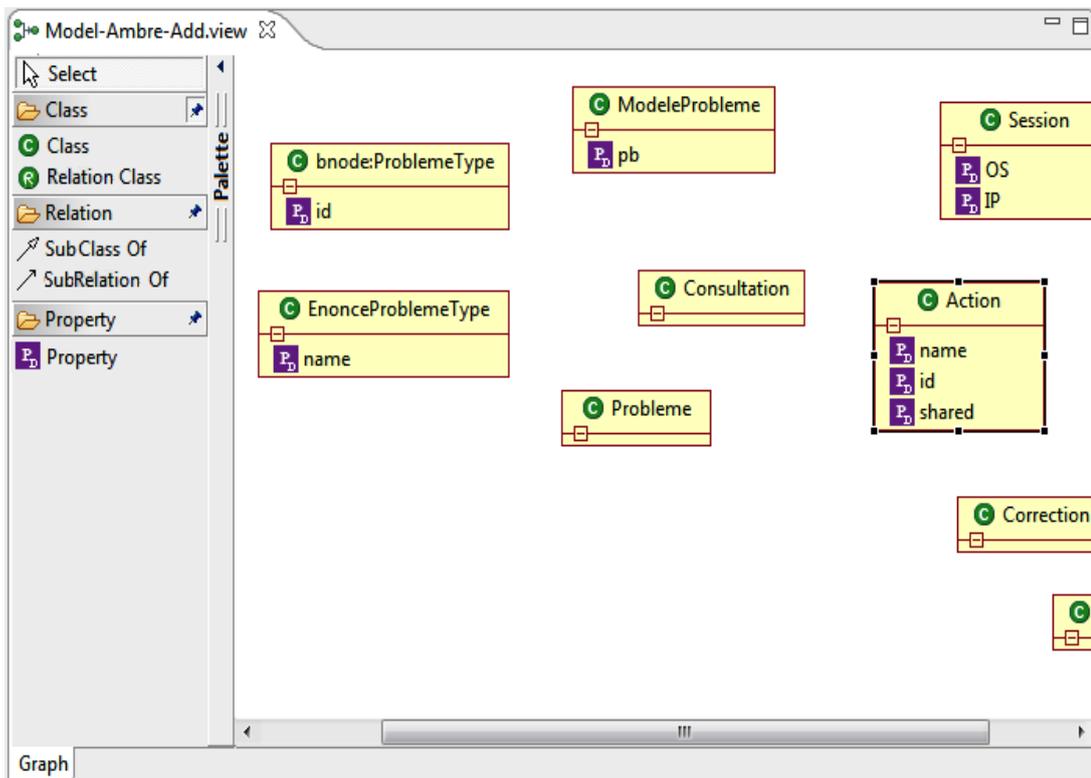


Fig.13 : La visualisation de modèle de traces dans ATER [Settoui, 11].

Les traces décrites par un modèle de trace ontologique sont constituées d'instances (individus) représentant leurs observés. La Figure (Fig.14) montre l'architecture du système ATER. L'architecture de la plateforme ATER distingue différents types d'utilisation du système ATER, qui correspondent aux flèches 1-5 de la Figure ci-dessous:

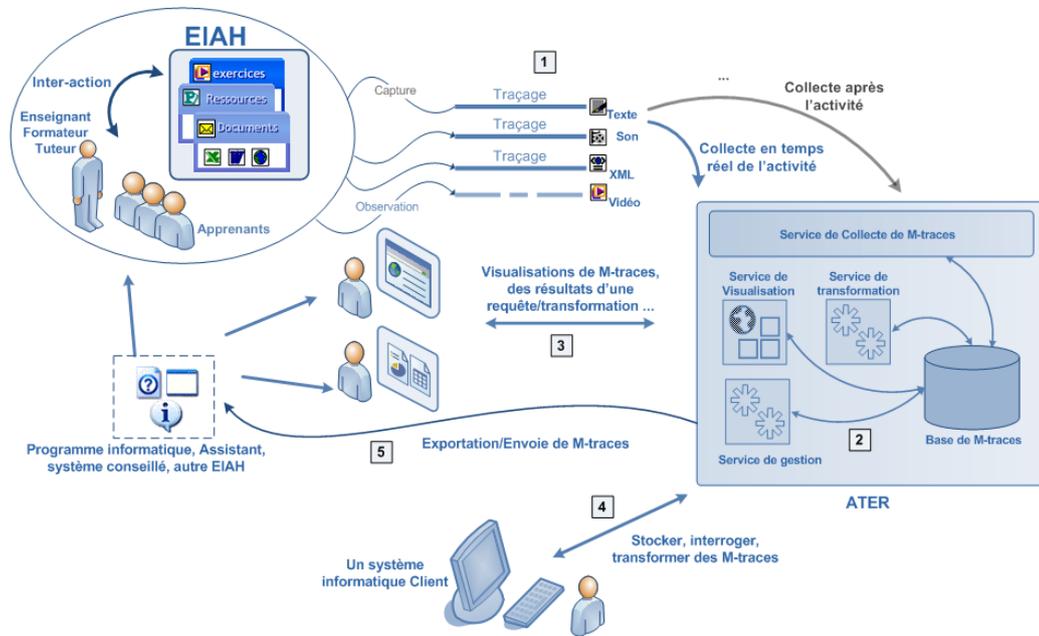


Fig.14 : Architecture globale du système ATER [Settoui, 11].

1. La collecte des traces à partir de sources de collecte (XML, texte, vidéo, etc.) avec un stockage sur le poste client. Le résultat de la collecte produit une nouvelle trace qui peut être traitée par le système ATER localement ou à distance. Ceci correspond par exemple au cas d'utilisation où chaque apprenant est tracé par l'EIAH utilisé localement et que l'enseignant peut se connecter à distance pour consulter cette trace.
2. L'utilisateur du système ATER peut gérer les traces, créer et gérer de nouveaux modèles et gérer les sources de collecte.
3. L'utilisateur du système ATER peut aussi visualiser une ou plusieurs traces. Il peut aussi visualiser les résultats d'une requête ou d'une transformation.
4. Un système informatique client du système ATER peut aussi stocker, interroger, transformer des traces et récupérer les résultats.
5. Le système ATER peut exporter et envoyer des M-Traces à une application tierce.

Enfin et comme pour tout système informatique, un administrateur est responsable de la gestion des utilisateurs et leurs droits d'accès.

4.2.10. Discussion

Les travaux sur MUSETTE portant sur la réutilisation d'expérience concernent des domaines assez variés : personnalisation de tableaux de bord pour la supervision de systèmes industriels, consultation de cours en ligne, conception assistée par ordinateur. Tous ces projets s'appuient sur la notion d'*épisode*, dont la notion de cas d'utilisation peut être vue comme une extension. D'autres travaux sur l'annotation de documents audiovisuels, dans le cadre du projet RECIS-SESAME, s'intéressent à la notion de modèle de tâche et de traces d'utilisation [Champin et al, 04].

Dans le projet EPICéA, l'évaluation du prototype intégrant le RÀPC comprendra deux étapes. Les auteurs de ce projet, ont estimé d'une part l'impact de l'intégration du RÀPC sur l'apprentissage humain. Cette évaluation s'appuiera sur la comparaison, en contexte expérimental écologique, des résultats d'apprentissage obtenus en utilisant soit e-Cursus, soit le prototype intégrant le RÀPC [Ollagnier, 03]. Les choix concernant le plan expérimental n'ont pas encore été faits. D'autre part, une évaluation cognitive et ergonomique de l'interface a été réalisée, qui se fera suivant deux axes : l'utilité et l'utilisabilité du prototype. L'utilité est l'adéquation entre les fonctions fournies par le système et celles nécessaires à l'utilisateur pour mener à bien les tâches qu'il doit remplir [Ollagnier, 03]. L'utilisabilité concerne l'adéquation entre la manière dont une tâche est réalisée par un utilisateur et le profil cognitif de cet utilisateur [Ollagnier, 03].

Dans Pixed, les enseignants peuvent représenter les connaissances du domaine dans le cadre d'une structure conceptuelle globale et définir des stratégies tutorielles, en termes de modes de parcours de cette structure, visant à permettre l'acquisition par l'apprenant de la connaissance ainsi représentée [Heraud, 04]. Pixed intègre des valeurs attachées aux nœuds du réseau dans son modèle d'apprentissage de l'apprenant.

Bien que la plateforme ATER soit fonctionnelle et permette d'offrir un certain nombre de services pour la gestion du cycle de vie des traces, des modèles et des ressources associés, elle n'est actuellement utilisée dans aucun projet. Ce projet de développement peut même être considéré comme un échec puisque aucun développement futur n'est prévu

que ce soit pour faire évoluer les fonctionnalités de la plateforme ou maintenir, tester et corriger son code.

Dans ATER, la collecte n'est pas générique et doit être codée spécifiquement pour chaque application. Elle peut être exécutée en temps réel ou en différé et permet de passer d'une ou plusieurs sources de collecte quelconques à une trace première stockée dans une base de traces RDF.

En effet, l'implémentation actuelle de la plateforme ATER est un prototype de recherche et est loin d'être un produit fini. Le prototype ATER présenté n'est toujours pas en version stable ou finale et certains modules ou fonctionnalités doivent être corrigés ou complétés. L'extension de la plateforme et la prise en main de son code source sont deux tâches fastidieuses.

5. Conclusion

Ce chapitre nous a permis, dans sa première section, de distinguer les profits de la réutilisation de connaissances en tant qu'un outil d'évolution des systèmes d'apprentissage à distances. Nous allons voir, dans la deuxième section du chapitre, comment contribuer à la réutilisation dans le domaine des ressources pédagogiques et plus particulièrement la réutilisation de traces d'apprentissage, afin d'exploiter ces derniers, parce qu'elles peuvent être des sources d'information importantes sur le processus d'apprentissage.

*Chapitre IV : La
conception du
système SRTn*

1. Introduction

Bon nombre d'EIAH tracent les interactions entre apprenants et environnement. Les traces d'interaction d'un apprenant donnent une image fidèle de son activité. Elles permettent par fois de refléter ce qu'il fait avec un EIAH lui permettant de revenir sur son activité et peut revoir les questions et les hypothèses qu'il a formulées tout au long de sa session d'apprentissage. Dans notre contribution, nous visons à lui permettre de consulter ou de suivre les traces similaires des apprenants qui ont réussi à atteindre leurs objectifs (réussir un test d'évaluation par exemple).

Dans ce chapitre, nous présentons notre travail qui concerne la réutilisation de traces d'apprentissage. Nous proposons dans ce sens, un système appelé : *Système de Réutilisation de Traces numériques (SRTn)*.

En premier lieu, nous décomposons les étapes de construction de notre système en deux grandes phases: *la phase de préparation* et *la phase d'exploitation*. La première inclut trois étapes : *collecte* de trace, *transformation de traces*, et indexation de la base de trace. La deuxième comporte deux étapes : mesure de similarité et visualisation de traces.

Nous allons détailler les deux phases ainsi que leurs étapes dans la suite de ce chapitre.

2. Architecture du système SRTn

Afin d'offrir une aide aux apprenants rencontrant des difficultés lors de la résolution des tests d'évaluation au cours de leur apprentissage dans une plateforme d'apprentissage en ligne, nous proposons un système qui réutilise les traces fournies par ceux-ci.

Nous allons voir dans cette section l'architecture générale et fonctionnelle du système de réutilisation des traces numériques (SRTn) et ses différents composants ainsi que leurs fonctionnalités.

2.1. Définitions

Avant d'entrer dans les détails des différents composants du système SRTn, nous devons d'abord définir les éléments clés exploités par ces composants.

a. TRACE PREMIERE

Une trace première est une observation ou un enregistrement de l'interaction de l'apprenant avec un système en vue d'une analyse.

b. Trace modélisée

Une trace modélisée est décrite par un modèle de traces et un ensemble d'instances de ce modèle, où chaque instance est située dans l'axe du temps, elle est générée par *le collecteur de traces*.

c. Trace de-modélisée

Une trace dé-modélisée est décrite par une structure compréhensible par l'apprenant (sous un format texte ou graphique), elle est générée par *le visualisateur de traces*.

d. Episode

Un épisode est un vecteur contenant des éléments bien définis de la trace. Il est construit à partir d'une phase de transformation des traces modélisées.

e. Modèle de traces

Un modèle de trace est une spécification formelle d'une structure définissant les éléments qui composent une trace modélisée.

2.2 Architecture générale du système SRTn

La figure (**Fig.15**) montre l'architecture générale du système SRTn. Le système est constitué, d'une manière ascendante, des composants et outils suivants :

- Un *collecteur de traces* : sert à collecter différentes sources de traçage (logs, BDD, fichier XML...etc.). Au niveau de cette étape, les *traces premières* collectées seront modélisées selon le *modèle de trace* proposé.
- Un *transformateur de traces*: sert à transformer les *traces modélisées* en *épisodes* à l'aide d'un *algorithme de transformation*.
- L'ensemble des épisodes obtenus, sera stocké dans une *base de connaissances*. Cette dernière est prête à recevoir aussi les *traces modélisées* et les *traces dé-modélisées*.
- Le système SRTn comporte aussi un autre composant qui est le *dé-modélisateur* de traces, sert à représenter les traces dans un format bien défini, afin de les visualiser aux apprenants.

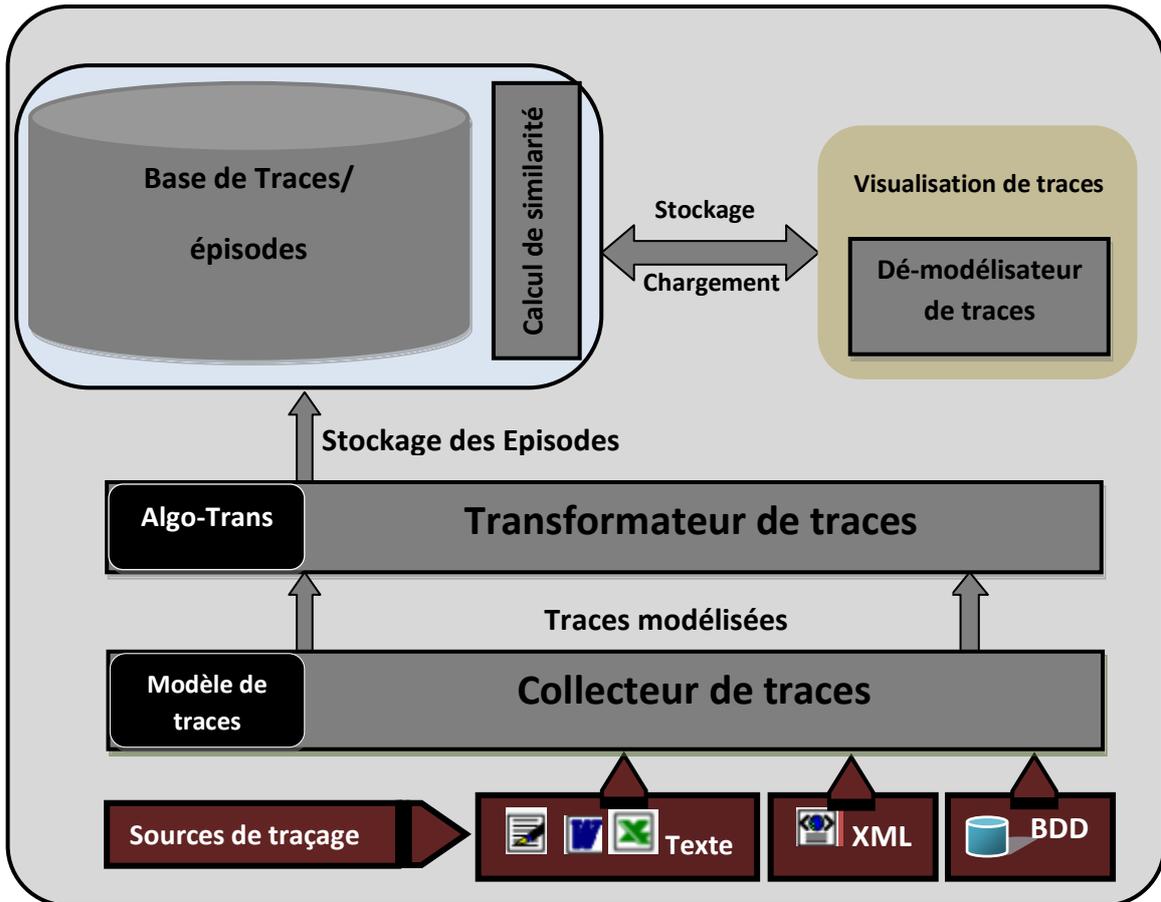


Fig.15 : Architecture générale du système SRTn.

2.3 Architecture fonctionnelle du système SRTn

La figure (Fig.16) montre l'architecture fonctionnelle du système SRTn. Cette architecture est constituée, comme nous avons mentionné précédemment, d'un processus de collecte (1) de traces premières, un système de transformation de traces modélisées (2) en épisodes (3), ces derniers sont enregistrés dans une base de traces (4).

Lors qu'un apprenant rencontre des problèmes d'évaluation, le système SRTn lui offre une aide, en lui fournissant des historiques d'autres apprenants (traces) dans le même contexte d'apprentissage.

A ce niveau là, et après avoir collecté et transformé les traces de cet apprenant demandeur de l'aide, SRTn fait appel à tous les épisodes stockés au niveau de la base des épisodes, et compare l'épisode en cours avec ceux-ci, à l'aide du mécanisme de *calcul de similarité*.

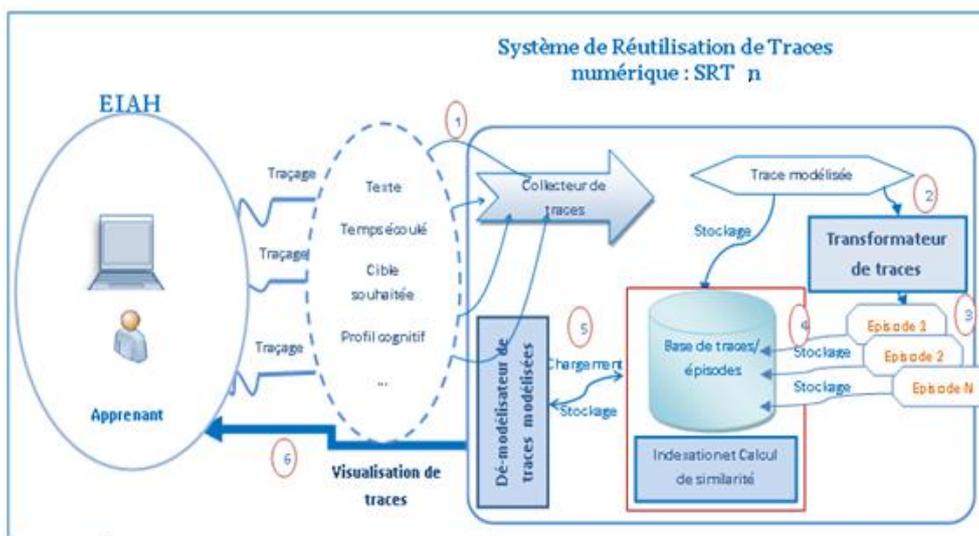


Fig.16 : Architecture fonctionnelle du système SRTn.

Une fois les/l'épisode(s) le(s) plus similaire(s) à l'épisode en cours est/sont trouvé(s), une requête sera envoyée à la base de traces afin d'extraire l'ensemble de traces à visualiser (6). Mais, avant de visualiser ces dernières, une autre étape qui est la dé-modélisation de traces (5), sera dé-modélisée ceci afin de les mettre sous une forme précise.

3. Phase de préparation

Nous présentons dans la figure (**Fig.17**), les différentes opérations effectuées pendant la phase de préparation. En effet, cette phase définit les éléments constituant le système SRTn, en d'autre terme c'est l'étape d'organisation des différents éléments du système élaboré.

3.1. Type de traces à collecter par le système SRTn

Le **Tableau.3** ci-dessous, illustre les différents types de traces pris en compte par le système SRTn.

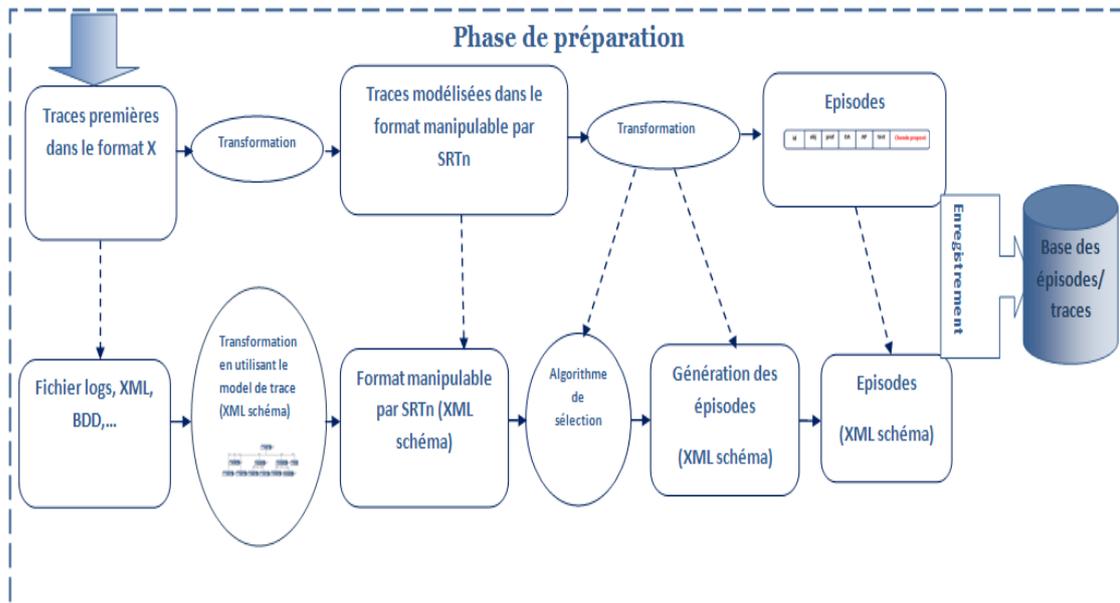


Fig.17: La phase de préparation.

Nous avons pris quatre types de traces numériques: **les traces informatives** (Informations personnelles, Informations techniques...), **les traces associées à l'exploitation des ressources** (Nom (référence) de la ressource traitée par un apprenant à un moment donné, Nombre d'accès à la ressource, Durée de consultation de la ressource, Historique du parcours des ressources, Nombre de pages de cours visitées...), **les traces liées à l'activité d'apprentissage** (Temps de réponse à une question, résultats de tests d'évaluation, détails sur les choix et les parcours de formation suivis...) et en fin **Traces liées à l'activité de communication** (Nombre de messages envoyés/reçus (mail), Message de communication (contenu), Demande d'aide en ligne, Demande d'assistance au tuteur, Nombre de messages postés/lus (forum)...).

Dans cette typologie, nous avons tout d'abord expliqué pourquoi choisir ces types de traces (i.e. les exploitations possibles des sous types de traces). Ensuite, nous avons présenté le but de réutilisation attendu de ces traces.

Les questions que nous allons poser sont : comment peut-on collecter ces traces ? Par quel moyen ? Et quelle est la forme qu'elles vont prendre (ces traces) après leur collecte ? Dans le reste de ce chapitre, nous donnerons des réponses aux questions précédentes.

	Nature de traces	Exploitations possibles et objectifs attendus
traces informatives	- Informations personnelles (âge, genre, ...)	- Comparaison à la situation d'apprenant en cours - Comparaison à la situation d'apprenant en cours
	- Informations techniques (IP, browser, SE, ...)	...
Traces associées à l'exploitation des ressources	- Nom (référence) de la ressource traité par un apprenant à un moment donné	- Informer et aiguiller l'apprenant en cours vers le bon choix... - Dégager des informations sur l'apprenant (capacités, compétences, ...)
	- Nombre d'accès	- Faire confiance au choix - Evaluer le niveau d'activité - Prendre la durée d'apprentissage en considération
	- Durée de consultation de la ressource (connexion)	- Dégager le cheminement conceptuel / profil d'apprenant - Informer et aiguiller l'apprenant en cours vers le bon choix
	- Historique du parcours des ressources	- Dégager le nombre de pages de cours consultées - Prendre la quantité d'apprentissage en considération
	- Nombre de pages de cours visitées	...
Traces liées à l'activité d'apprentissage et d'évaluation	- Résultats de tests d'évaluation	- Evaluation de l'assimilation de concepts/connaissances - Connaître le succès / échec de l'apprenant.
	- Détails sur les choix et les parcours de formation suivis	- Suivi du travail de l'apprenant / Respect du scénario. - donner une idée sur son efficacité, faiblesse par rapport aux autres apprenants
	- Temps de réponse à une question	...
Traces liées à l'activité de communication	- Nombre messages envoyés/reçus (mail)	- les apprenants qui ont un nombre de contacts avec le tuteur et/ou l'enseignant sont ceux qui ont obtenu les meilleurs résultats
	- Message de communication (contenu)	- l'exploitation des discussions évalue la qualité des interactions
	- Demande d'aide en ligne	...
	- Demande d'assistance aide au tuteur	...
	- Nombre messages postés/lus (forum)	...

Tableau.3 : Typologie de traces utilisées par le système SRTn.

3.2. Collecte et filtrage de traces

La première étape dans **SRTn** est la collecte et le filtrage des traces numériques. Dans notre contexte, une *trace numérique* est issue de l'observation d'une activité (**Fig.18**), elle

représente une signature d'un processus interactionnel d'un apprenant. L'objectif de cette étape est de dresser une liste de traces numériques effectivement collectées sur divers systèmes. Le filtrage des traces est une étape qui sert à ne sélectionner que les traces numériques fixées dans le **tableau.4**.

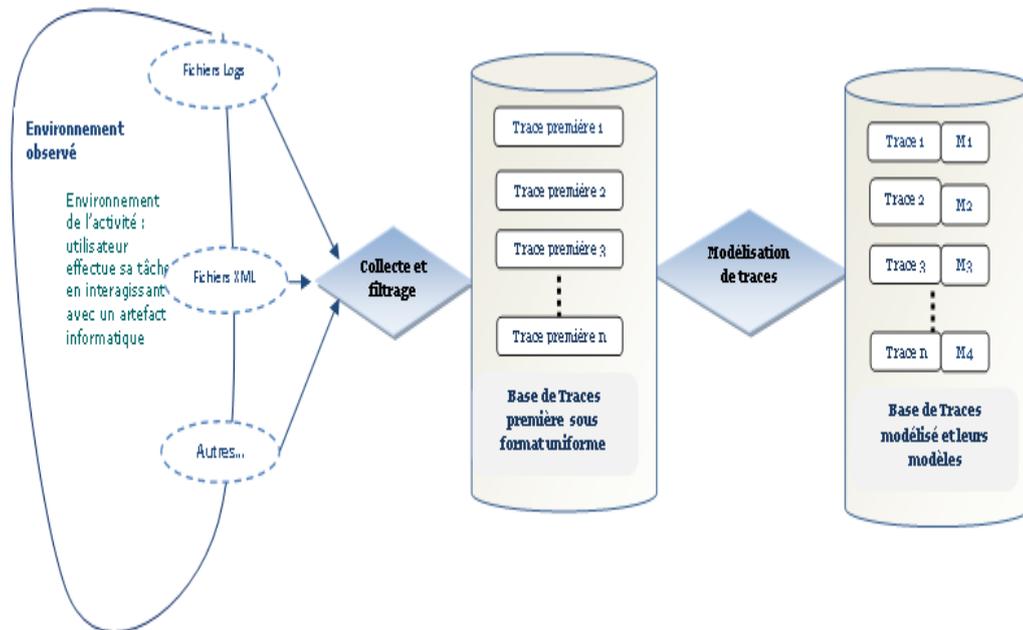


Fig.18 : collecte, filtrage et modélisation de traces.

a) Listes des traces à collecter

Dans le **Tableau.4**, nous présentons la liste des traces, appelées traces premières par la suite. La liste retenue n'est bien sur pas exhaustive, mais reflète les principales traces retenues dans les systèmes d'apprentissage (**Fig.19**).

b) Pourquoi utiliser des traces modélisées ?

SRTn est conçu pour collecter la plus part des actions de l'apprenant. Contrairement aux traces de plus bas niveau (e.g. frappes clavier et cliques de souris), habituellement collectées par certains systèmes, la granularité des traces collectées par **SRTn** est beaucoup plus pertinente et permet souvent d'obtenir des traces ayant une sémantique de plus haut niveau grâce à une étape de filtrage.

Type de traces	Indication de la trace
Traces informatives	<ul style="list-style-type: none"> - Information personnelle : Nom/prénom de l'apprenant, âge de l'apprenant, e-mail de l'apprenant... - information techniques : l'adresse IP, browser, système d'exploitation...
Traces associées à l'exploitation des ressources	<ul style="list-style-type: none"> - Texte étudié - Temps de l'étude de ce texte - Nombre d'accès à ce texte - Nombre d'accès au système - Date de connexion au système - Duré de la connexion au système -Nom/référence de la ressource traitée par l'apprenant à un moment donné - Duré de consultation de la ressource - Nombre de pages visitées - Historique des pages visitées
Traces liée à l'activité d'apprentissage et d'évaluation	<ul style="list-style-type: none"> - Détails sur les choix et les parcours de formation suivis - Nombre d'exercices résolus - Nombre d'exercice sans réponses - Résultat du test + évaluation (note, moyenne, numéro d'exercice, nombre des réponses justes, nombre des réponses fausses pour chaque exercice) - Temps de réponse à une question - Marques de réussite à des tests et questionnaires, actions effectuées sur le système
Trace liées à l'activité de communication	<ul style="list-style-type: none"> - Nombre de messages envoyés (E-mail) - Nombre de messages reçu (E-mail) - Fréquence, volume et destinataire de messages - Date de l'envoi du message - Demande d'assistance au tuteur - message de communication (contenu)

Tableau.4 : Traces numériques collectées par SRTn.

Nous avons pensé dès le début autour de la notion de traces modélisées pour **SRTn**. En effet, l'un des intérêts les plus importants de la modélisation de traces est la simplicité de la réutilisation de celles-ci.

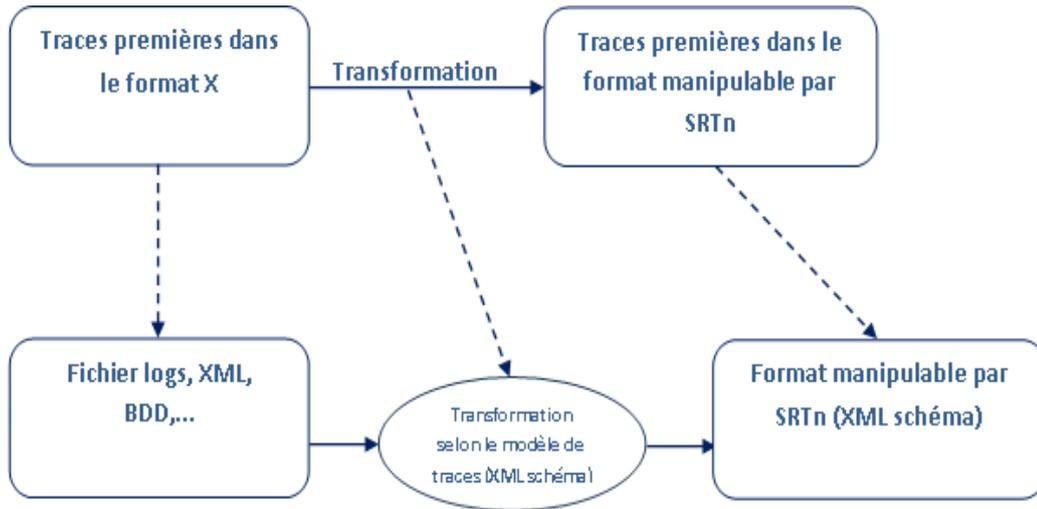


Fig.19 : Diagramme de collecte de traces.

La figure (Fig.20) représente le modèle de trace exploité par SRTn. Nous avons proposé un modèle de trace adopté aux modèles de traces que nous avons retrouvé dans la plus part des travaux de recherche étudiés. Le modèle de trace ainsi défini doit nous permettre de pouvoir traiter les traces les plus fréquemment rencontrées dans le cadre des situations d'apprentissage.

La modélisation d'une trace donc, permet la description de nombreux formats de traces, depuis des fichiers textes structurés jusqu'aux bases de données.

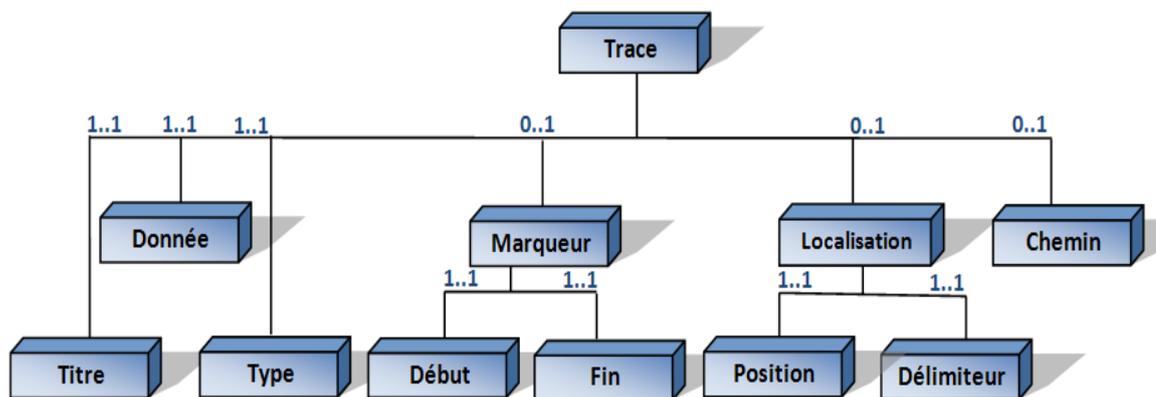


Fig.20: Le modèle d'une trace.

Une trace se décompose de :

- Le champ 'Donnée', qui est collectée automatiquement par le dispositif d'apprentissage sur l'activité d'un apprenant de la session d'apprentissage, comme le temps passé à lire une page, ou le nom d'une page lue, ou encore le résultat d'un exercice...

- Le 'titre' d'une trace désigne le contenu et lui associe un sens (par exemple : "date de début d'activité", "réponse au QCM"...).
- Le 'type' peut prendre pour valeur "Texte", "XML" ou "BDD". (Cette liste est ouverte).
- Le 'chemin' contient le chemin d'accès à la donnée, pour une trace de type "BDD" ou "XML" (soit une requête XPath ou SQL par exemple).

En ce qui concerne les fichiers de "logs" sous forme de fichiers texte structurés, un ensemble de champs est proposé, permettant d'indiquer la localisation de la donnée dans une chaîne de caractères, soit par la position des caractères, soit en utilisant des marqueurs.

Des attributs spécifiques ont été prévus :

- 'Début' donne la position du premier caractère du contenu.
- 'Fin' donne la position du dernier caractère.
- 'Délimiteur' correspond au délimiteur utilisé pour séparer la chaîne de caractères en différentes sections.
- 'Position' donne la position de la section de chaîne de caractères à conserver.

3.3 Processus de transformation

Le processus de transformation de traces premières (Fig.21) sert à reformuler ces dernières en des autres appelées : *épisodes*. En effet ce processus est un Schéma XML permettant de représenter les épisodes, selon un algorithme de recherche dans le modèle de traces pour en extraire les différents éléments de l'épisode.

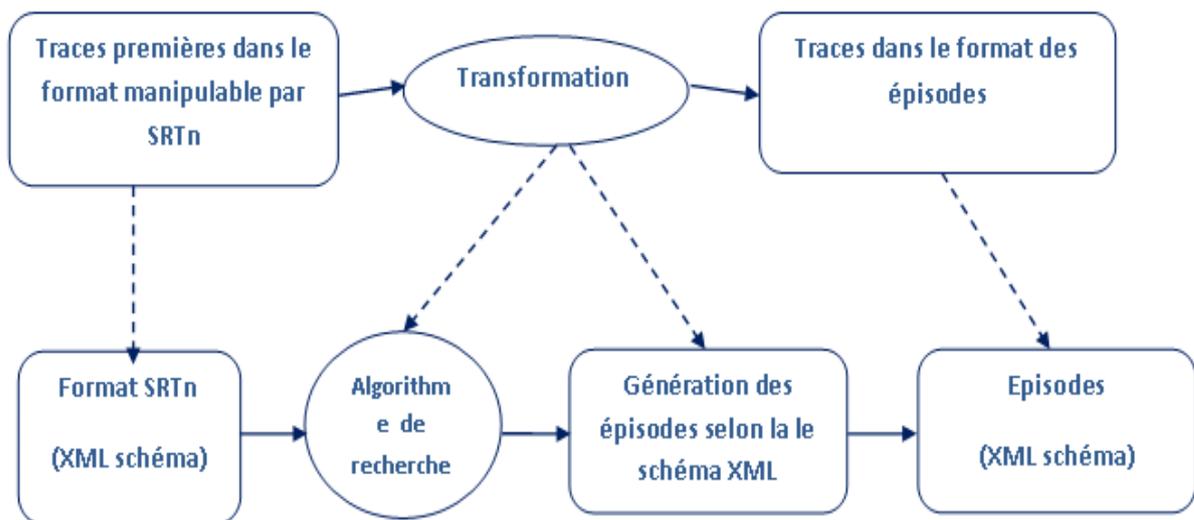


Fig.21: Diagramme de transformation de traces.

3.3.1 Structure générale d'un épisode

Un épisode est un vecteur constitué de plusieurs champs (**Fig.22**). Il est constitué des éléments (ou bien champs) suivants :

- **id** : **id**entifiant, de type numérique pour identifier les épisodes.
- **obj** : est l'objectif d'apprentissage, représente le degré de maîtrise de l'objet d'apprentissage associé.
- **OA** : la liste des **Objets** (l'objet) d'**Apprentissage** visitée par un apprenant avant la résolution des tests d'évaluation. (Ensemble d'objets d'apprentissage, bien défini par l'auteur (enseignant) OAi)
- **AP** : **Activité d'aP**prentissage : chaque objet contient un ou plusieurs activités d'apprentissage mises en œuvre par l'enseignant (Ensemble des activités d'apprentissage bien défini par le concepteur : APi) et effectué par les apprenants; Par exemple : consultation d'un chapitre d'un cours, exécute un test...
- **prof** : Profil cognitif de l'apprenant. (Excellent, très bien, bien, moyen ou faible)
- **test** : un *test d'évaluation* qui peut être un QCM,...
- **chemin proposé** : Chemin des traces à visualiser dans la base de traces démodélisées.



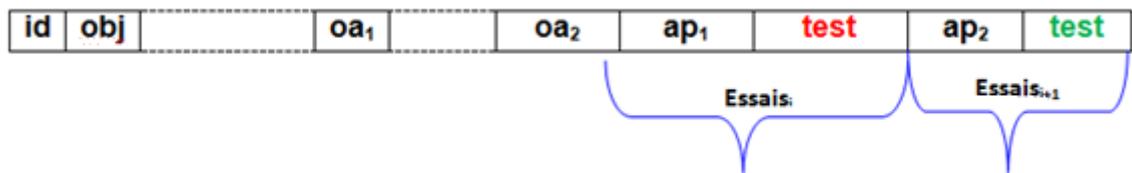
Fig.22 : Structure générale d'un épisode.

Les éléments précédents sont organisés dans l'épisode de la façon suivante :

- Un épisode contient un est un seule objet d'apprentissage, car un apprenant ne peut accéder à un cours que par son accès à l'objet qui le présente.
- Un épisode peut contenir plus qu'une activité d'apprentissage, et plus qu'un *test d'évaluation*. Le test peut être validé juste ou non (selon son score).
- Un épisode se termine toujours par le champ '*chemin proposé*' qui donne l'adresse des traces à visualiser dans la base.
- Une activité d'apprentissage ou un test d'évaluation sont toujours précédé par d'autres éléments de l'épisode dont au moins un objet d'apprentissage.

- Un test d'évaluation est une activité d'apprentissage ayant un statut particulier dans l'épisode.
- On dit qu'un *test d'évaluation* est validé juste, lorsque sa valeur « score » est supérieure ou égale à un certain pourcentage, et faux dans le cas contraire.
- La valeur du champ '*obj*' (objectif d'apprentissage) de type alphabétique, peut contenir deux valeurs « *atteint* » ou « *non-atteint* ».
- La valeur du champ '*prof*' (profil cognitif de l'apprenant) de type numérique, prend en compte cinq valeurs (*excellent, très bien, bien, moyen et faible*). On associe à chaque type une valeur numérique fixe (5,4,3,2,1 respectivement) pour simplifier le calcul de similarité entre deux profils cognitifs.
- Un essai dans un épisode commence toujours par une activité d'apprentissage (*AP*) et se termine par un *test d'évaluation*. Un essai est précédé soit par un objet d'apprentissage (*OA*) ou un *test d'évaluation*.

▪ Exemple d'un essai



Cet exemple illustratif d'un essai, montre deux occurrences d'essais pour l'objet (*oa₂*), l'un avec un test validé faux, l'autre avec un test validé juste pour un apprenant donné.

3.3.2. Structure formelle d'un épisode

La structure formelle des épisodes est sous forme de balise XML schéma, dont les éléments d'un épisode sont organisés selon la forme XML suivante :

```
<base-episodes>
<episode id/>
<objet-apprentissage name/>
<activité-apprentissage name/>
<test name/>
```

```
</episode>
```

```
</base-episodes>
```

Le schéma XML d'un épisode est alors :

```
<?xml version= "1.0" ?>
```

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
```

```
<xs:element name="obj" type="xs:string"/>
```

```
<xs:element name="prof" type="xs:string"/>
```

```
<xs:element name="chemin" type="xs:url"/>
```

```
<xs:attribut name="id" type=="xs:ID"/>
```

```
<xs:attribut name="score" type=="xs:integer"/>
```

```
<xs:element name="base-episodes">
```

```
<xs:complexType>
```

```
<xs:sequence>
```

```
<xs:element ref="episode" maxoccurs="unbounded"/>
```

```
</xs:sequence>
```

```
</xs:complexType>
```

```
</xs:element>
```

```
<xs:element name=" episode">
```

```
<xs:complexType>
```

```
<xs:sequence>
```

```
<xs:element ref="OA" minoccurs="1" maxoccurs="unbounded"/>
```

```
<xs:element ref="AP" minoccurs="0" maxoccurs="unbounded"/>
```

```
<xs:element ref="test" minoccurs="0" maxoccurs="unbounded"/>
```

```
</xs:sequence>  
  
</xs:complexType>  
  
</xs:element>  
  
<xs:element name=" OA">  
  
<xs:complexType>  
  
<xs:sequence>  
  
<xs:attribut ref="name"/>  
  
</xs:sequence>  
  
</xs:complexType>  
  
</xs:element>  
  
<xs:element name=" AP">  
  
<xs:complexType>  
  
<xs:sequence>  
  
<xs:attribut ref="name"/>  
  
</xs:sequence>  
  
</xs:complexType>  
  
</xs:element>  
  
<xs:element name=" test">  
  
<xs:complexType>  
  
<xs:sequence>  
  
<xs:attribut ref="name"/>  
  
<xs:attribut ref="score"/>  
  
</xs:sequence>
```

</xs:complexType>

</xs:element>

</xs:schema>

Après la transformation des traces collectées, nous sommes devant d'autres problèmes. Comment indexer notre base d'épisodes? Et comment calculer la similarité entre ces épisodes ?

Dans ce qui suit, nous allons expliquer la solution, que nous avons élaborée, pour répondre aux deux questions précédentes.

3.4. Indexation

L'indexation d'épisodes dans la base des épisodes se fait en se basant sur l'indexation vectorielle.

Nous créons une table d'index classique, contenant comme entrée un *test*, et comme sortie une liste d'épisodes incluant des validations justes pour ce *test*.

Nous justifions le choix du *test d'évaluation* comme index de notre base d'épisodes par le besoin de l'apprenant à être guidé par le système au bon chemin de traces, qui lui permet de réussir son évaluation. En d'autres termes, le champ 'test' dans un épisode offre la possibilité d'extraire, lors d'une recherche, tous les épisodes contenant les tests d'évaluation similaires au test d'évaluation en cours, ainsi que le score du test ne permettant de filtrer que les épisodes ayant les tests validés juste.

L'exemple dans la figure (**Fig.23**) suivante montre les différents composants de la table d'index :

N°	Terme	Liste
01	test1	
02	test2	
⋮	⋮	⋮

Fig.23 : Différents composants de la table d'index.

Pratiquement, le *numéro de l'épisode* (N° d'épisode) représente le (*id*) de celui-ci. La liste peut être représentée par une balise XML.

4. Phase d'exploitation

Nous présentons dans la figure (Fig.24), les différentes opérations effectuées pendant la phase d'exploitation. En effet cette phase comporte les éléments qui permettent l'utilisation des opérations offertes par le système SRTn.



Fig.24 : La phase d'exploitation.

4.1. Mesures de similarité

Afin de comparer entre les épisodes, nous utilisons deux étapes pour mesurer la similarité entre eux. La première se base sur la table d'index et la deuxième se base sur les éléments constituant l'épisode.

4.1.1 Calcul de similarité entre épisodes base sur la table d'index

Le premier calcul de similarité entre les épisodes dans la base se fait à l'aide de la table d'index, précisément il se base sur l'élément *poids* du test.

Lorsqu'une demande d'aide est lancée par l'apprenant, la procédure de recherche sur l'épisode le plus similaire à l'épisode en cours se commence. Ensuite, l'algorithme de recherche sélectionne tous les épisodes contenant le *test d'évaluation* à rechercher. Le choix des épisodes les plus similaires dépend du *poids* du *test* dans cet épisode, ce qui implique que les épisodes les plus similaires à l'épisode en cours sont ceux qui ont le *poids* du *test* le plus fort.

Nous exploitons la formule qui calcule le poids du terme, dans notre cas en remplaçant un terme par un *test*, un document par un épisode et un corpus par une base des épisodes.

La formule (1) du poids du terme sera adaptée donc comme suit :

$$\mathbf{Poids} = \mathbf{QF} * \mathbf{IEF} \dots (1)$$

Avec: **QF** est QCM Frequency (la fréquence du QCM dans l'épisode), **IEF** est Inverted Episode Frequency (la fréquence de l'épisode renversé).

La fréquence du *test* dans un épisode se calcule comme suit :

$$\mathbf{Qf}_{i,j} = \text{freq}_{i,j} / \max_k \text{freq}_{k,j}$$

Avec : $\text{freq}_{i,j}$: est le nombre d'occurrences du *test* q_i dans l'épisode e_j , et $\max_k \text{freq}_{k,j}$ est le nombre d'occurrences du *test le plus fréquent* dans l'épisode e_j .

La discriminance d'un *test* dans un épisode est calculé par la formule (2) suivante :

$$\mathbf{lef}_i = \log(N/n_i) \dots (2)$$

Où N est le nombre total des épisodes dans la base et n_i le nombre des épisodes dans lesquels le *test* q_i apparaît. Lorsque ce *test* est apparu dans beaucoup d'épisodes, la valeur Ief_i est petite. Elle appartient à l'intervalle $[0, \log(N)]$.

Après la sélection des épisodes ayant des tests d'évaluation similaires, un filtrage selon le score du test se fait, afin de ne choisir que les épisodes avec un test validé juste.

4.1.2 CALCUL DE SIMILARITE ENTRE EPISODES BASE SUR LES DISTANCES ENTRE SES ELEMENTS

Cette deuxième étape de calcul de similarité vient pour éviter le problème de conflits qui peut arriver après le premier calcul détaillé dans la section précédente, ainsi, pour donner des résultats plus pertinents (réduire la quantité de formation le minimum possible).

Cette opération se base notamment sur le calcul de distance entre les éléments identiques des épisodes, précisément : objets d'apprentissage, profils cognitifs des apprenants, essais et enfin les épisodes eux mêmes.

a) DISTANCE ENTRE OBJETS D'APPRENTISSAGE

Dans la plus part des EIAH, les objets d'apprentissage ont des relations entre eux. Généralement, cette relation est une mesure de '*propagation*' saisie par un enseignant /tuteur. Cette valeur '*de propagation*' propage un objet d'apprentissage (OA_i) aux autres objets d'apprentissage constituant le cours.

La distance entre deux objets d'apprentissage est calculée, en utilisant la distance euclidienne, comme suit (formule (3)):

Soit OA_1 , OA_2 deux objets d'apprentissage,

$$D(OA_1, OA_2) = \sqrt{\text{val_propagation}(OA_1, OA_2)^2 + \text{val_propagation}(OA_2, OA_1)^2} \dots (3)$$

Cette distance est donc égale à :

- 1 s'il n'existe aucune relation entre OA_1 et OA_2 ,
- 0 si OA_1 apporte 100% de maîtrise à OA_2 et que OA_2 apporte 100% de maîtrise à OA_1 .

b) Distance entre les profils cognitifs des apprenants

Le champ profil cognitif (*prof*) est représenté par sa valeur, ce champ peut prendre cinq valeurs : Excellent (5), très bien (4), bien (3), moyen(2) et faible (1), chaque valeur est représentée par un numéro pour simplifier le calcul de similarité.

- si la valeur absolue de la différence entre deux profils est égale à 4 le résultat sera profils Très différents,
- si la valeur absolue de la différence entre deux profils est égale à 3 le résultat sera profils différents,
- si la valeur absolue de la différence entre deux profils est égale à 2 le résultat sera profils moyennement similaire,
- si la valeur absolue de la différence entre deux profils est égale à 1 le résultat sera profils similaire.
- si la valeur absolue de la différence entre deux profils est égale à 0 le résultat sera profils même-profil.

c) Distance entre essais

La formule (4) utilisée pour calculer la distance entre essais, elle est inspirée et adaptée à partir d'une formule utilisée par J.Héraud dans [Héraud, 03].

$$D(E_1, E_2) = d(OA_1, OA_2) * (1 + |nbAp_1 - nbAp_2|) \dots (4)$$

Avec E_1, E_2 sont deux essais, $d(E_1, E_2)$ est la distance entre les deux essais, OA_1, OA_2 sont deux objets d'apprentissage de E_1 et E_2 respectivement.

$nbAp_x$: est le nombre des activité d'apprentissage dans l'essai E_x .

d) Distance entre épisodes

La formule (5) utilisée pour calculer la distance entre épisodes, elle est inspirée et adaptée à partir d'une formule utilisée par J.Héraud dans [Héraud, 03].

$$D(X_1, X_2) = D(\text{prof}_1, \text{prof}_2) * \frac{\sum_{j=M}^{i=N} (E_{1,i}, E_{2,j})}{nbE_{min}} \dots (5)$$

Avec: X_1 et X_2 sont deux épisodes,

prof_i : est le profil cognitif associé à l'apprenant approprié à l'épisode X_i .

e) Heuristique Pour Le Calcul De Distance Entre Episodes

Lorsqu'un apprenant demande l'aide du système, afin de résoudre son QCM, le système *SRTn* sélectionne les traces évidentes pour lui. Il les transforme, par la suite, en épisode appelé : *épisode cible*, ce dernier est décomposé en traces, puis en essais. La distance entre cet *épisode cible* et les épisodes de la base se calcule selon l'heuristique suivant :

- Pour $i=1..n$ faire (n : est le nombre d'épisodes dans la base de cas)

Extraire tous les épisodes qui ont $\text{poids}_{\text{qcm}} \geq \text{seuil}$, tel que *seuil* est la valeur minimum prendre en considération par le système, pour dire que cet épisode est similaire à l'épisode cible.

- A ce niveau, le système commence le calcul de la distance:

Si $D(\text{prof}_{\text{cible}}, \text{prof}_{\text{source}}) \ll$ alors

Calculer $D(\text{OA}_{\text{cible}}, \text{OA}_{\text{source}})$

Si $D(\text{OA}_{\text{cible}}, \text{OA}_{\text{source}}) \ll$ alors

Calculer $D(\text{E}_{\text{cible}}, \text{E}_{\text{source}})$

Si $D(\text{E}_{\text{cible}}, \text{E}_{\text{source}}) \ll$ alors

Calculer $D(\text{X}_{\text{cible}}, \text{X}_{\text{source}})$.

Le choix de l'épisode le plus similaire repose sur la plus petite valeur de distance $D(\text{X}_{\text{cible}}, \text{X}_{\text{source}})$.

L'opérateur « \ll » signifié « plus petit à un seuil bien défini »

f) Algorithme de recherche des épisodes

L'algorithme de recherche des épisodes se lance lorsqu'un apprenant demande l'aide du système, lors de son évaluation. Cet algorithme se résume en quelques étapes :

- Accéder à la table d'index ;
- Extraire la liste d'épisodes similaires en se basant sur le poids du test le plus fort ;
- Filtrer la liste selon le score du test d'évaluation (Sélectionner que les tests validés justes) ;

- Accéder à la base des épisodes ;
- Extraire tous les épisodes filtrés ;
- Calculer la similarité entre ceux-ci et l'épisode en cours ;
- Garder juste les épisodes les plus similaires à l'épisode en cours.

4.2 Visualisation des traces

La visualisation des traces consiste pour un utilisateur à explorer une trace ou une base de traces, de façon plus au moins interactive.

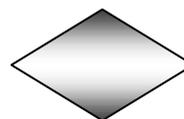
La visualisation des traces d'apprentissage, notamment les traces numériques (traces d'évaluation, traces de communication et traces d'apprentissage), permet aux apprenants de voir l'historique des activités d'autres apprenants. Le système **SRTn** offre un outil de visualisation des traces, par exemple : nom de la ressource traitée par un apprenant à un moment donné, nombre d'accès à la ressource, durée de consultation de la ressource, historique du parcours des ressources, nombre de pages de cours visitées, temps de réponse à une question, résultats de tests d'évaluation, détails sur les choix et les parcours de formation suivis...etc.

- La représentation des traces à visualiser dans SRTn (Dé-modélisation de traces)

Le système SRTn offre une visualisation graphique de traces (**Fig.25**), chaque type de traces possède sa propre représentation sur un axe de temps, dont le début et la fin de la trace est marquée par un petit cercle et une flèche respectivement, et les différents types de traces chacun par un signe spécifique. La trace à visualiser sera extraite depuis le modèle de trace qui la concerne, c'est-à-dire, nous allons construire la trace à visualiser pour un apprenant à partir du modèle de trace (dé-modélisation), rappelons que nous n'allons pas visualiser la trace collectée telle quelle.

* Traces d'apprentissage

Ce type de traces est représenté par un losange.



* Traces d'évaluation

Ce type de traces est représenté par un carré.



* Traces de communication



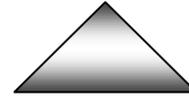
Ce type de traces est représenté par une ellipse.

* Traces associées à l'exploitation des ressources



Ce type de traces est représenté par un rectangle.

* Traces informatives



Ce type de traces est représenté par un triangle.

Exemple d'une trace visualisée :

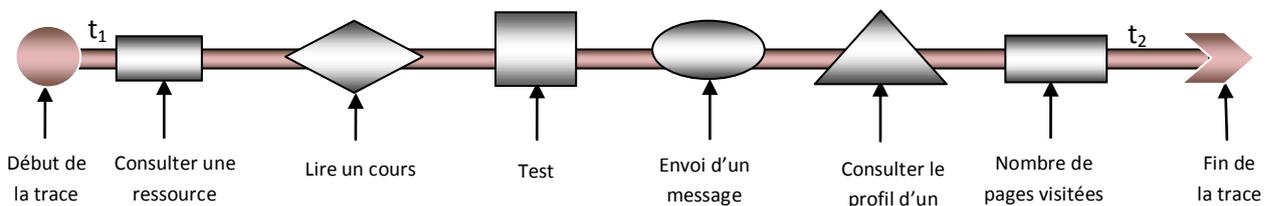


Fig.25 : Exemple d'une trace visualisée.

5. Modem de traces

Nous avons lancé le nom *modem* à deux étapes de traitement de traces: modélisation de traces collectées et la visualisation de ceux-ci comme il est illustré dans la figure (Fig.26), où le modem est entouré par une zone mauve.

L'utilité de ce modem de traces est de simplifier deux tâches primordiales dans notre travail :

5.1 La réutilisation des traces d'apprentissage collectées

La *modélisation* des traces donne une structure uniforme aux différents types de traces collectées, ce qui facilite leur exploitation et précisément leur réutilisation ultérieure.

5.2 La visualisation de traces

Ou la *dé-modélisation* de traces, qui donne la possibilité de présenter les traces dans un format claire et compréhensible avec les apprenants (axe de temps), toute en les construisant à partir du modèle de traces approprié.

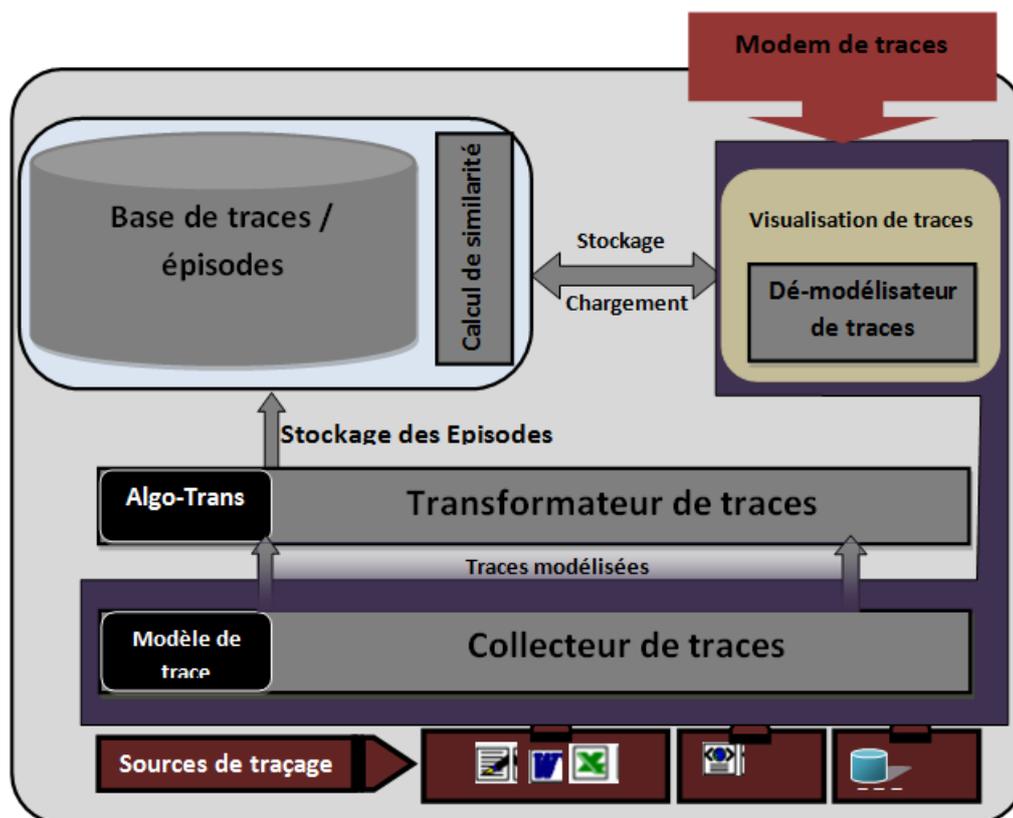


Fig.26 : Modem de traces.

5.3 Les avantages du modem de traces

Le modem de traces offre plusieurs avantages pour le système SRTn et l'apprenant en même temps, dont nous allons citer quelques uns :

- Il traite n'importe quel type de traces premières et les uniformiser en un seul modèle de traces.
- Il facilite l'exploitation des traces par le transformateur de SRTn, lors de la construction des épisodes.
- Il facilite la consultation et simplifie la compréhension de traces par un apprenant demandant l'aide du système lors d'une session d'apprentissage, (rapprocher la représentation des traces à la logique de réflexion des apprenants (axe de temps)).

6. Conclusion

Dans ce chapitre, nous avons exposé notre formalisme du système de réutilisation de traces numériques (SRTn).

SRTn vise à assister l'apprenant aux cours de son interaction avec un EIAH, en lui offrant la possibilité de réutiliser des traces des autres apprenants ayant des comportements similaires à cet apprenant. Cette réutilisation lui permet de réduire le temps de réflexion à résoudre son test d'évaluation.

La construction du système SRTn comporte deux grandes phases importantes : la phase de préparation et la phase d'exploitation. La première phase inclut trois étapes : collecte, transformation de traces premières en épisodes et indexation de la base de traces. La deuxième comprend deux étapes: calcul de similarité entre les épisodes et visualisation des traces. Nous avons aussi donné l'utilité et les avantages du *modem de traces* qui constitue un composant essentiel du système SRTn.

*Chapitre V : La
validation et
l'implémentation du
prototype*

1. Introduction

Une fois la conception du système SRTn est achevée, nous sommes passés à une nouvelle étape, en l'occurrence, la validation, laquelle consiste à tester la fiabilité d'une phase très intéressante dans le système, qui est le calcul de similarité entre épisodes

Nous montrons dans ce chapitre une première validation de notre système proposé. Il consiste à donner des exemples sur la sélection de l'épisode le plus similaire à l'épisode en cours, par le calcul de similarité entre les épisodes dans la base de traces et celui de l'apprenant en cours.

A la fin de ce chapitre, nous présenterons les résultats obtenus, nos observations et remarques.

2. Exemple illustratif

2.1 Collecte et transformation de traces

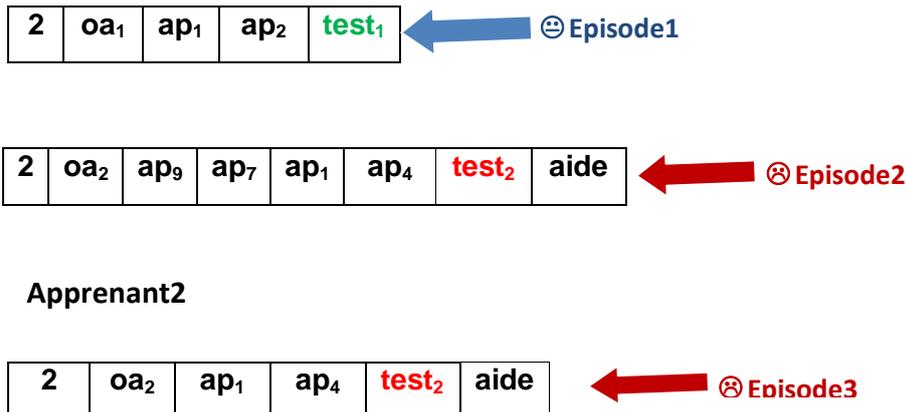
Dans cet exemple nous avons montré comment le système collecte et transforme les traces de deux apprenants devant un EIAH intégrant le système SRTn.

La figure (**Fig.28**) décrit trois épisodes de deux apprenants différents, apprennent dans le même EIAH. Nous remarquons, d'après la figure, que le premier apprenant (**apprenant 1**), ayant un profil cognitif égal à deux ($prof_1 = 2$). Il consulte l'objet (oa_1), réalise deux activités d'apprentissage (ap_1, ap_2) et valide en juste le test d'évaluation ($test_1$). Ensuite, il consulte l'objet (oa_2), réalise quatre activités d'apprentissage (ap_9, ap_7, ap_1, ap_4), valide en faux le test d'évaluation ($test_2$). A ce niveau là, l'**apprenant 1** demande l'aide du système. Le deuxième apprenant (**apprenant 2**) ayant un profil cognitif égal aussi à deux ($prof_2 = 2$). Il consulte l'objet (oa_2), réalise deux activités d'apprentissage (ap_1, ap_4) et valide en faux le test d'évaluation ($test_2$). Et là, il demande aussi de l'aide pour qu'il puisse valider son test en juste.

Les trois épisodes illustrés dans la figure (**Fig.27**), sont obtenus après la réalisation des deux étapes à savoir, la collecte et la transformation de traces premières. Dans la première étape, le système SRTn sélectionne les traces des deux apprenants selon le modèle de traces. Les traces ainsi obtenues, sont transformées à l'aide du schéma de transformation, afin d'extraire les trois épisodes.

Remarquant que nous n'avons pas attaché le champ « *chemin proposé* » à la fin de l'épisode, et cela juste pour simplifier la compréhension de cet exemple applicatif.

Apprenant1



Apprenant2

Fig.27 : Exemple de construction des épisodes.

2.1 Extraction des épisodes pertinents a partir de la base de traces/épisodes

Après les deux étapes collecte et transformation de traces en épisodes, le système SRTn cherche dans la base de traces/épisodes, les épisodes les plus pertinents, en se basant sur deux instances de calcul de similarité, l'une se base sur la table d'index et l'autre sur des formules qui calculent la distance entre les épisodes obtenus (du premier calcul de similarité) et les deux épisodes en cours.

a) Calcul de similarité en se basant sur la table d'index de la base

La figure (Fig.28), exprime l'ensemble des épisodes utilisés pour expliquer notre approche. En plus des deux épisodes en cours précédemment décrits, nous avons utilisé d'autres épisodes. Après le premier calcul de similarité, qui consiste à ne sélectionner que les épisodes ayant les poids des « tests » les plus forts (i.e. les tests qui sont similaires aux tests des épisodes en cours), ensuite, le système ne choisit que les tests similaires et les valide justes aussi, en se basant, pour ce dernier critère, sur le score des tests.

id	prot	EPISODES																
?	2	oa ₂	ap ₇	ap ₉	ap ₁	ap ₄	test ₂	aide										
?	2	oa ₂	ap ₁	ap ₄	test ₂	aide												
Episodes extraits à partir de la base																		
o1	3	oa ₂	ap ₁	ap ₂	test ₂	ap ₂	ap ₅	ap ₈										
o2	5	oa ₂	ap ₂	ap ₁	ap ₃	ap ₉	test ₁											
o3	4	oa ₃	ap ₂	ap ₁	test ₂	ap ₅	ap ₁	ap ₂	test ₂	ap ₃	ap ₄	test ₃						
o4	1	oa ₂	ap ₂	ap ₁	ap ₄	ap ₂	test ₂	ap ₇	ap ₆	ap ₈	test ₄							
o5	4	oa ₃	ap ₉	ap ₃	ap ₇	test ₂	ap ₂	ap ₁	ap ₂	test ₂	ap ₁	ap ₂	ap ₄	test ₄	ap ₅	ap ₆	ap ₈	test ₄

Fig.28 : Les épisodes en cours et les épisodes extraits à partir la base de traces.

b) Calcul de similarité en se basant sur le calcul de distances entre les épisodes en cours et les épisodes conclus

Dans le **Tableau.5**, montre le calcul des différentes distances entre les deux épisodes en cours et les épisodes conclus de la base à partir du premier calcul de similarité. Nous pouvons constater que le tableau est décomposé en quatre parties. De gauche à droite, La première partie du tableau (la plus à gauche), contient le premier l'épisode du premier apprenant (en haut) et également, les résultats du calcul de distances entre l'objet d'apprentissage de l'épisode du *premier apprenant* et les différents objets d'apprentissage des épisodes de la base, de même que les distances entre ce dernier et les différents essais illustrés dans la deuxième partie du tableau.

La deuxième partie du tableau, figure les différents essais des épisodes de la base. Ensuite, la troisième partie du tableau, présente les résultats du calcul de distances entre l'objet d'apprentissage de l'épisode du *deuxième apprenant* et les différents objets d'apprentissage des épisodes de la base (On a Supposé que la valeur de propagation proposée par le concepteur de l'EIAH est égale à 1. La distance entre l'objet et lui-même est par défaut égale à 0.), de même que les distances entre ce dernier et les différents essais illustrés dans la deuxième partie du tableau.

id	prof	Episode 1 en cours						
		oa ₁	ap ₁	ap ₂	ap ₃	ap ₄	test ₁	
1	2	1	1					
2	1	1	2					
3	1	1	1					
	1	1	1					
	1	1	1					
4	1	1	2					
	1	1	2					
5	1	1	1					
	0	0						
	1	1	1					
	1	1	1					

prof	Ensemble des Episodes et essais						
	oa ₁	ap ₁	ap ₂	ap ₃	ap ₄	test ₁	
3	oa ₁	ap ₁	ap ₂	test ₁			2 essais
3	ap ₁	ap ₂	ap ₃				
5	oa ₁	ap ₁	ap ₂	ap ₃	ap ₄	test ₁	1 essai
4	oa ₁	ap ₁	ap ₂	test ₁			
4	ap ₁	ap ₂	ap ₃	test ₁			3 essais
4	ap ₁	ap ₂	test ₁				
1	oa ₁	ap ₁	ap ₂	ap ₃	ap ₄	test ₁	
1	ap ₁	ap ₂	ap ₃	test ₁			2 essais
4	oa ₁	ap ₁	ap ₂	ap ₃	ap ₄	test ₁	
4	ap ₁	ap ₂	ap ₃	test ₁			4 essais
4	ap ₁	ap ₂	ap ₃	test ₁			
4	ap ₁	ap ₂	ap ₃	test ₁			

prof	Episode 2 en cours			
	oa ₁	ap ₁	ap ₂	test ₁
2	0	0		
0	0	0		
0	0	0		
1	1	1		
0	0	0		
0	0	0		
0	0	0		
1	1	2		
1	1	1		
1	1	1		
0	0	0		
1	1	1		

D(profi, profj)	D(X ₁ , X _i)	D(X ₂ , X _j)
1	3	0.5
3	6	1
2	1	5
1	0.75	1
2	1	5

D(OA _i , OA _j)		D(E _i , E _j)	
---------------------------------------	--	-------------------------------------	--

D(OA _i , OA _j)		D(E _i , E _j)	
---------------------------------------	--	-------------------------------------	--

Tableau.5 : calcul de similarité entre deux épisodes en cours pour le même test d'évaluation.

Finalement, la quatrième partie du tableau montre respectivement, les résultats de distances entre les profils cognitifs des deux *apprenants* (l'*apprenant 1* et l'*apprenant 2* ont le même profil cognitif), et les résultats de distances entre les différents épisodes de la base et les deux épisodes en cours.

Les épisodes les plus similaires aux épisodes en cours, sont ceux qui ont la plus petite valeur de distance. D'après le tableau, l'épisode identifié à '4' est le plus similaire à l'épisode approprié à « *apprenant 1* », de même que, l'épisode identifié à '2' est le plus similaire à l'épisode approprié à « *apprenant 2* ».

Il ne reste qu'à visualiser l'ensemble de traces associées à chaque épisode pour l'apprenant approprié, selon le modèle de visualisation graphique proposé.

3. Implémentation du prototype

3.1 Présentation des outils de développement

3.1.1 Microsoft VISUAL STUDIO

Microsoft Visual Studio est une suite de logiciels de développement pour Windows conçue par Microsoft. La dernière version s'appelle **Visual Studio 2010 (Fig.29)**.

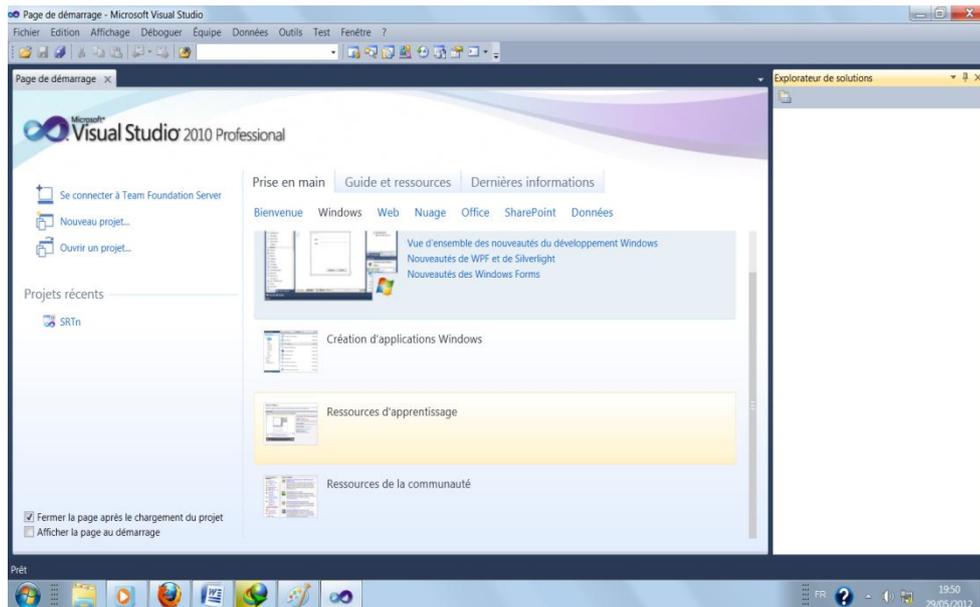


Fig.29 : La page de démarrage de Visual studio 2010.

Visual Studio est un ensemble complet d'outils de développement permettant de générer des applications Web ASP.NET, des Services Web XML, des applications bureautiques et des applications mobiles. Visual Basic, Visual C++, Visual C# et Visual J# utilisent tous le même environnement de développement intégré (IDE, Integrated Development Environment), qui leur permet de partager des outils et facilite la création de solutions faisant appel à plusieurs langages. Par ailleurs, ces langages permettent de mieux tirer parti des fonctionnalités du Framework .NET, qui fournit un accès à des technologies clés simplifiant le développement d'applications Web ASP et de Services Web XML grâce à Visual Web Developer.

3.1.2 Wampserver

WampServer (anciennement **WAMP5**) est une plateforme de développement Web de type WAMP, permettant de faire fonctionner localement (sans se connecter à un serveur externe) des scripts PHP (**Fig.30**). WampServer n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs (Apache et MySQL), un interpréteur de script (PHP), ainsi que phpMyAdmin pour l'administration Web des bases MySQL.

Il dispose d'une interface d'administration permettant de gérer et d'administrer ses serveurs au travers d'un *tray icon* (icône près de l'horloge de Windows).

La grande nouveauté de WampServer 2 réside dans la possibilité d'y installer et d'utiliser n'importe quelle version de PHP, Apache ou MySQL en un clic. Ainsi, chaque développeur peut reproduire fidèlement son serveur de production sur sa machine locale.

Le 20 février 2011 est sortie la version 2.2d. Cette version intègre Apache 2.2.21, MySQL 5.5.20, PHP 5.3.10, XDebug 2.1.2, XDC 1.5, PhpMyadmin 3.4.10.1, SQLBuddy 1.3.3, webGrind 1.0.

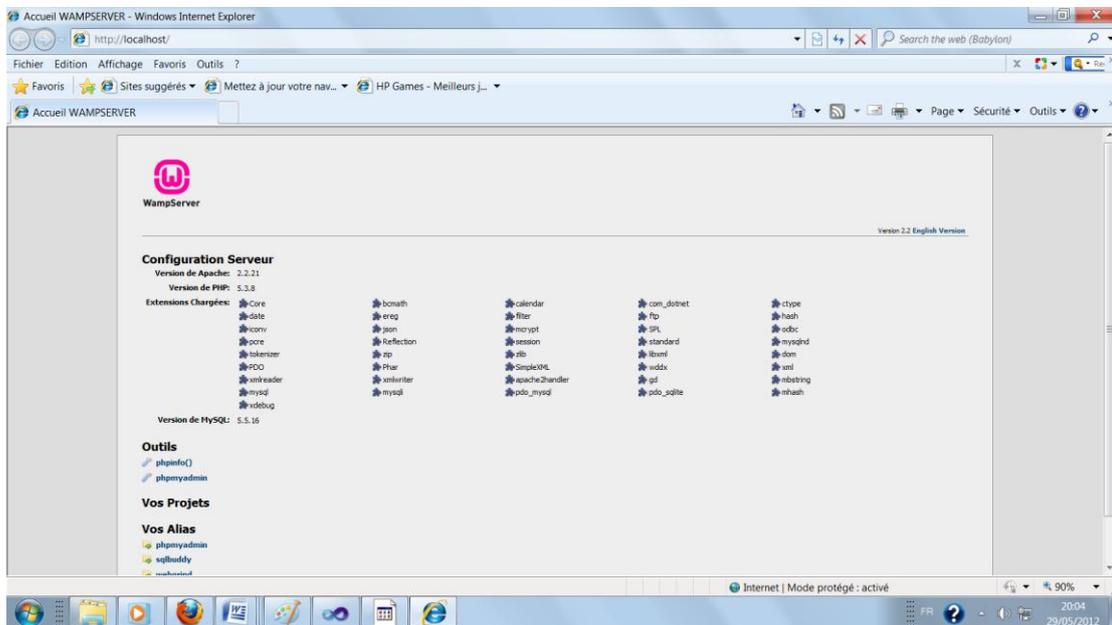


Fig. 30: Accueil WAMPSEVER.

3.1.3 PhpMyAdmin

PhpMyAdmin (PMA) est une application Web de gestion pour les systèmes de gestion de base de données MySQL réalisée en PHP et distribuée sous licence GNU GPL.

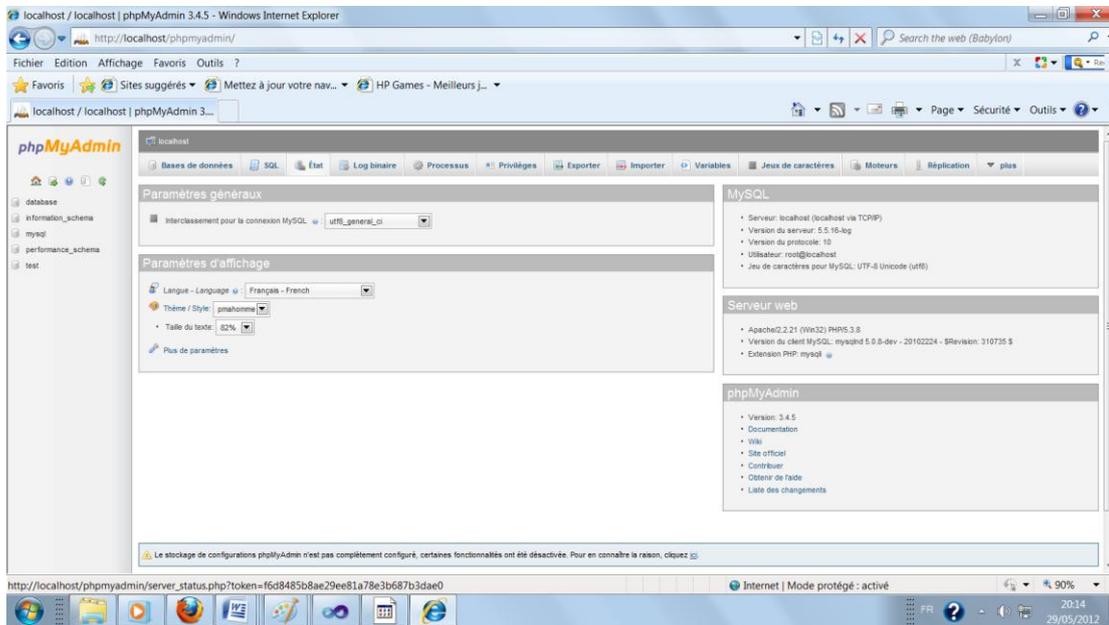


Fig. 31: Interface de PhpMyAdmin3.4.5.

Il s'agit de l'une des plus célèbres interfaces pour gérer une base de données MySQL sur un serveur PHP (**Fig.31**). De nombreux hébergeurs, qu'ils soient gratuits ou payants, le proposent ce qui permet à l'utilisateur de ne pas avoir à l'installer.

Cette interface pratique permet d'exécuter, très facilement et sans grandes connaissances dans le domaine des bases de données, de nombreuses requêtes comme les créations de table de données, les insertions, les mises à jour, les suppressions, les modifications de structure de la base de données. Ce système est très pratique pour sauvegarder une base de données sous forme de fichier .sql et ainsi transférer facilement ses données. De plus celui-ci accepte la formulation de requêtes SQL directement en langage SQL, cela permet de tester ses requêtes par exemple lors de la création d'un site et ainsi de gagner un temps précieux.

3.2 Objectifs attendus

Afin de valider les formules que nous avons proposé, (atteindre l'épisode le plus similaire à l'épisode en cours), nous avons mis en évidence un prototype de calcul des distances entre ces épisodes.

3.3 Validation de l'approche proposée

Comme nous avons signalé précédemment, nous avons utilisé PhpMyAdmin pour créer notre base de données grâce à sa simplicité. Cette base de données est structurée sous forme de 3 tables (**Fig.32**) comme suit :

Table	Action	Lignes	Type	Interclassement	Taille	Perte
episodeindexed	Afficher Structure Rechercher Insérer Vider Supprimer	0	InnoDB	latin1_swedish_ci	16,0 KiB	-
episodes	Afficher Structure Rechercher Insérer Vider Supprimer	5	InnoDB	latin1_swedish_ci	16,0 KiB	-
index	Afficher Structure Rechercher Insérer Vider Supprimer	2	InnoDB	latin1_swedish_ci	16,0 KiB	-
3 tables Somme		7	InnoDB	latin1_swedish_ci	48,0 KiB	0

Fig. 32: Base des épisodes.

- episodeindexed
- episodes
- index

3.4 Présentation des interfaces du prototype implémenté

L'ensemble des formes suivantes, montrent les résultats de calcul de similarité entre les épisodes en cours, et les épisodes enregistrés dans la base. Comme nous remarquons, chaque épisode en cours (en bas de la forme) est comparé par un nombre des épisodes extraits de la base (Listes des épisodes sélectionnées de la base), la plus petite valeur du champ 'DE', de chaque ensemble d'épisode, concerne l'épisode le plus similaire à l'épisode en cours.

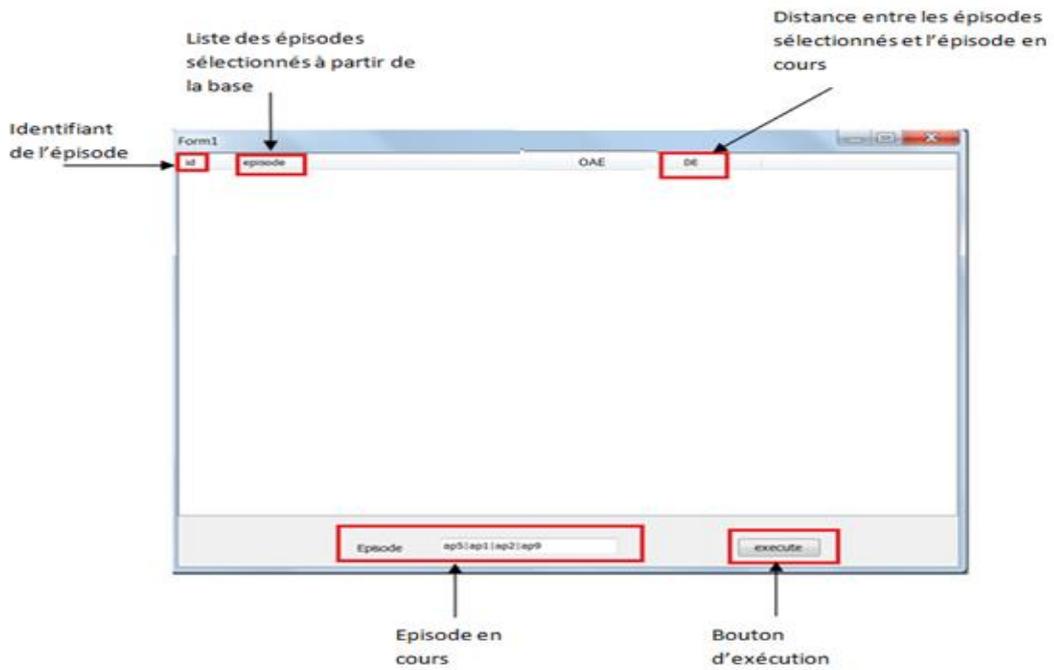


Fig. 33: interface de calcul de similarité.

id	episode	OAE	DE
1	ap1 ap2 test2	0	0
2	ap5 ap6 test2	1	2
3	ap2 ap1 ap3 ap9 test2	1	2
4	ap5 ap1 test2	0	0
5	ap1 ap2 test2	0	0
6	p4 test2	1	3
7	ap3 ap1 ap4 ap2 test2	1	2
8	ap6 ap8 test4	1	2
9	ap9 ap5 ap7 test5	1	1
10	ap1 ap5 test2	0	0
11	ap2 ap4 test4	1	2
12	ap6 ap8 test4	1	2

Fig. 34: Exemple 1.

The screenshot shows a window titled "Form1" containing a table with the following data:

id	episode	OAE	DE
1	ap1 ap2 test2	1	2
2	ap5 ap6 test2	1	2
3	ap2 ap1 ap3 ap9 test2	1	2
4	ap5 ap1 test2	1	2
5	ap1 ap2 test2	1	2
6	p4 test2	1	3
7	ap3 ap1 ap4 ap2 test2	1	2
8	ap6 ap8 test4	1	2
9	ap9 ap5 ap7 test5	1	1
10	ap1 ap5 test2	1	2
11	ap2 ap4 test4	1	2
12	ap6 ap8 test4	1	2

Below the table, there is a label "Episode" followed by a text input field containing "oa2|ap1|ap4|test2" and an "execute" button.

Fig.35: Exemple 2.

The screenshot shows a window titled "Form1" containing a table with the following data:

id	episode	OAE	DE
1	ap1 ap2 test2	1	2
2	ap5 ap6 test2	1	2
3	ap2 ap1 ap3 ap9 test2	1	2
4	ap5 ap1 test2	1	2
5	ap1 ap2 test2	1	2
6	p4 test2	1	3
7	ap3 ap1 ap4 ap2 test2	1	2
8	ap6 ap8 test4	1	2
9	ap9 ap5 ap7 test5	1	1
10	ap1 ap5 test2	1	2
11	ap2 ap4 test4	1	2
12	ap6 ap8 test4	1	2

Below the table, there is a label "Episode" followed by a text input field containing "ap9|ap7|ap4|test3" and an "execute" button.

Fig. 36: Exemple 3.

4. Conclusion

Nous avons essayé de présenter à travers des exemples théoriques et d'autres pratiques, la principale fonctionnalité réalisée par le système (calcul de similarité entre épisodes). Cependant, il est important de mentionner que ce prototype est incomplet (nous avons implémenté la partie qui concerne le calcul de similarité entre les épisodes).

A travers ce prototype, nous pouvons dire que notre approche est efficace et réalisable. La partie développée montre bien la faisabilité de nos formules de calcul de similarité (l'une de nos contributions majeures dans ce mémoire). L'exemple choisi contient des tâches qui peuvent correspondre réellement à un cas réel (en fait, nous l'avons pris des traces d'utilisation d'un système EIAH développé à l'université de Guelma). Il nous reste à implémenter les autres composants du système et le mettre en exploitation dans une situation réelle.

Conclusion générale

Le travail de recherche présenté dans ce mémoire s'inscrit dans le domaine des environnements informatiques d'apprentissage humain (EIAH). Notre recherche porte plus particulièrement sur : La généralisation d'une aide individualisée pour les apprenants. L'approche choisie pour aider les utilisateurs ou plus particulièrement les apprenant est la réutilisation de traces numériques qui consiste à construire, mémoriser et interpréter des épisodes similaires afin de proposer une séquence d'étapes adaptées (visualisation de traces pertinentes) dans le cadre de l'épisode en cours.

L'objectif principal de ce travail est que la réutilisation de traces des apprenants passées doit permettre de guider l'apprenant en cours, dans sa navigation dans le cours, et plus particulièrement dans son évaluation.

Nous décomposerons les étapes de construction de notre système en deux grandes phases:

- *La phase de préparation.*
- *la phase d'exploitation.*

La première phase contenant deux composants : le *collecteur* de traces et le *transformateur* de traces, avec une étape d'indexation de la base de traces. La deuxième comporte une étape de mesure de similarité et un seule composant : le visualisateur de traces.

Nous avons présenté donc, un état de l'art sur les traces d'apprentissage, leurs types, traitements, utilisations, et plus particulièrement leurs réutilisations.

Dans notre contribution, nous sommes engagés à réaliser deux buts essentiels :

- comment choisir les traces les plus similaires à la situation de l'apprenant en cours (qui demande l'aide du système SRTn).
- le deuxième est comment visualiser ces traces pour faciliter la compréhension de l'apprenant.

Pour cela, nous avons suggéré un modèle de traces pour faciliter la réutilisation des traces, le modèle proposé est inspiré des modèles de traces que nous avons les retrouvé et étudié dans la plus part des travaux existants.

Nous avons proposé aussi une structure bien définie de l'épisode, extraite du modèle de traces. Les épisodes sont indexés et mémorisés dans une base de traces. Pour cela, nous avons utilisé une table d'index en se basant sur l'indexation vectorielle, qui index nos épisodes selon les tests d'évaluation et leurs scores.

Ensuite, nous avons lancé le « *modem* » de traces, qui est le responsable de la modélisation des traces collectées, et de la visualisation des traces pertinentes à la situation en cours. Ce modem de traces, comporte alors deux composants essentiels, le modélisateur de traces (selon un nouveau modèle de traces), et un dé-modélisateur de traces (selon la visualisation proposée).

Nous achevons notre recherche par une validation de notre proposition. En mettant des exemples pour tester les différentes formules et algorithmes proposés.

Les perspectives de ce travail s'orientent vers :

- ✓ L'implémentation des différents composants du système SRTn proposé.
- ✓ L'utilisation d'un système de recommandation qui s'occupera de la réutilisation de traces.
- ✓ La contribution du mécanisme de raisonnement à partir de cas, où un épisode sera considéré comme étant un cas.
- ✓ Offre d'autres possibilités d'aide à l'apprenant (par exemple lui offre de l'aide dans son apprentissage,... etc.) ainsi que pour l'enseignant, en lui fournissant des outils de suivi à partir de la réutilisation des traces de ces apprenants.

Bibliographie

BIBLIOGRAPHIE

[Avouris et al, 07]

N.AVOURIS, A. DIMITRACOPOULOU, V.KOMIS, M. MARGARITIS, Beyond logging of fingertip actions: analysis of collaborative learning using multiple sources of data. To appear in Journal of Interactive Learning Research JILR, vol. 18(2). (2007).

[Balacheff et al, 97]

N.BALACHEFF, M.BARON, C.DESMOULINS, M.GRANDBASTIEN ET M.VIVET : Conception d'Environnements Interactifs d'Apprentissage avec Ordinateur. Tendances et perspectives. In : journées nationales du PRCIA, Grenoble, France. (1997).

[Baker et al, 03]

M.J.BAKER, M.QUIGNARD, K. LUND et A.SÉJOURNÉ, Computer-supported collaborative learning in the space of debate. In Proceedings of the International Conference on Computer Support for Collaborative Learning 2003, p. 11-20. (2003).

[Benayache, 05]

A. BENAYACHE, Construction d'une mémoire organisationnelle de formation et évaluation dans un contexte e-learning : le projet MEMORAE », Université de Technologie de Compiègne — UTC — Spécialité : Technologies de l'Information et des Systèmes (T.I.S.). (2005).

[Bratitsis et Dimitracopoulou, 05]

T.BRATITSIS, A. DIMITRACOPOULOU, Data Recording and Usage Interaction Analysis in Asynchronous Discussions: The D.I.A.S. System. Workshop on Usage Analysis in Learning Systems, International Conference on Artificial Intelligence in Education AIED. (2005).

[Broisin et Vidal, 07]

J.BROISIN, P.VIDAL, Une approche conduite par les modèles pour le traçage des activités des utilisateurs dans des EIAH hétérogènes, *Revue STICEF*, Volume 14, ISSN : 1764-7223, mis en ligne le 13/03/2008, <http://sticef.org>. (2007).

[Champin et al, 04]

P-A.CHAMPIN, Y.PRIE, A.MILLE, MUSETTE: a Framework for Knowledge Capture from Experience. EGC'04, Clermont Ferrand. (2004).

[Chikh, 04]

A.CHIKH, Une méthodologie de réutilisation en ingénierie du document : Le système, Thèse de Doctorat d'état en Informatique, INI. (2004).

[Choquet et Iksal, 08]

C. CHOQUET, S. IKSAL, Modélisation et construction de traces d'utilisation d'une activité d'apprentissage : une approche langage pour la réingénierie d'un EIAH, Revue STICEF, Volume 14, ISSN : 1764-7223, mis en ligne le 07/03/2008, <http://sticef.org>. (2007).

[Delestre et al, 98]

N.DELESTRE, B.RUMPLER. Architecture d'un serveur multimédia pour les sciences de l'ingénieur. In *NTICF'98*, pages 383–391. (1998).

[Delestre et al, 99]

N.DELESTRE , W.AZIZ et S.BARTHÉLÉMY, How to share multimedia data about engineering sciences , In *CAEE'99*, pages 72–78. (1999).

[Després, 01]

C.DESPRÉS, Modélisation et Conception d'un Environnement de Suivi Pédagogique Synchrone d'Activités d'Apprentissage à Distance. Thèse Doctorat. Université du Maine. (2001).

[Djouad, 08]

T.DJOUAD, Analyser l'activité d'apprentissage collaboratif : Une approche par transformations spécialisées de traces d'interactions, 2nde Rencontres Jeunes Chercheurs en EIAH, RJC-EIAH'2008. (2008).

[Egyed-Zsigmond et al, 03]

E.EGYED-ZSIGMOND, A.Mille et Y. PRIE. “Club ♣ (Trèfle): A Use Trace Model.” P. 1065 dans *Case-Based Reasoning Research and Development*, vol. 2689/2003, Lecture Notes in Computer Science. Springer Berlin / Heidelberg. <http://www.springerlink.com/content/c75rmtdpblbplwa6/>. (2003).

[Fischer, 05]

FABIEN FISCHER L'UTILISATION DES TRACES NUMÉRIQUES DANS L'ENSEIGNEMENT À DISTANCE, Mars 2005.

[Gwenegan, 05]

R. GWENEGAN, Structuration et analyse de traces hybrides issues de situation d'apprentissage. Rapport de Master 2, de l'Université Josef Fourier, Grenoble Alpes. (2004-2005).

[Guéraud, 04]

V.GUÉRAUD, J.-M.ADAM, J.-P. PERNIN, G.CALVARY et J.-P. DAVID, L'exploitation d'Objets Pédagogiques Interactifs à distance : le projet FORMID. STICEF, vol. 11, ISSN: 1764-7223, www.sticef.org. (2004).

[Hathaway et al, 97]

J.HATHAWAY, S.KIRSHBAUM, The reusable content object strategy. Technical report, Oracle. (1997).

[Hardy et al, 04]

J.HARDY, M.ANTONIOLETTI, S.BATES, e-Learner Tracking : Tools for discovering Learner Behavior. *The IASTED International Conference on Web-base Education, Innsbruck, Austria.* (2004).

[Heraud, 04]

J-M HERAUD, PIXED : Une approche collaborative de l'expérience et l'expertise pour guider l'adaptation des hypermédias. Thèse de doctorat de L'UNIVERSITÉ LYON I, (2004).

[Hightower et al, 98]

R.HIGHTOWER, L.RING, J.HELFMAN, B.BEDERSON et J.HOLLAN, Graphical multiscale web histories: A study of PadPrints. Proc. of Hypertext '98, p. 58-65. (1998).

[Iksal et Choquet, 05a]

S.IKSAL, C.CHOQUET, An open architecture for usage analysis in a e-learning context. *The 5th IEEE International Conference on Advanced Learning Technologies, Kaohsiung, Taiwan,* p. 177-181. (2005).

[Jermann, 04]

P.JERMANN, Computer Support for Interaction Regulation in Collaborative Problem Solving, Thèse de doctorat en informatique, Genève. (2004).

[Labat, 02]

J.M. LABAT, EIAH : Quel retour d'informations pour le tuteur ? *Colloque Technologies de l'Information et de la Connaissance dans l'Enseignement supérieur et l'industrie (TICE'02)*, p 81-88. (2002).

[Laflaquiere et al, 05]

J.LAFLAQUIERE, P-A.CHAMPIN, Y.PRIE et A.MILLE, Approche de modélisation de l'expérience d'utilisation de systèmes complexes pour l'assistance aux tâches de veille informatiquement médiées. Conférence : a représentation orientée utilisation des informations (la représentation de l'information selon les usages potentiels). (2005).

[Laflaquiere et Prié, 07]

J. LAFLAQUIERE, Y.PRIE, Des traces modélisées, un nouveau support pédagogique, in Learning Object Repositories Research Network (LORNET 2007), Montréal. (2007).

[Le Calvez et al, 03]

F.LE CALVEZ, H.GIROIRE, J. DUMA, G.TISSEAU, M.URTASUN, Combien?
a Software to Teach Students How to Solve Combinatorics Exercises. *Workshop "Advanced Technologies for Mathematics Education" AIED 2003*, p. 447-454. (2003).

[Maurice, 98]

M.MAURICE-DEMOURIoux, Une méthodologie d'aide à la conception de systèmes d'information fondé sur la réutilisation, Thèse de doctorat en informatique, Université de Paris-Dauphine. (1998).

[Mazza et Dimitrova, 03]

R.MAZZA, V. DIMITROVA, CourseVis: Externalising Student Information to Facilitate Instructors in Distance Learning. *AIED 2003*. p. 279-286. (2003).

[Mechid, 08]

S.MECHID, Conception des scénarios d'apprentissage par réutilisation des patrons, Mémoire

En vue de l'obtention du diplôme de Magister en Informatique.(2008)

[Mcilory, 76]

M.D.MCILORY, Mass-produced software components. *Software Engineering Concepts and Techniques* (1968 NATO Conference on Software Engineering). In J.M Buxton, P. Naur, and B. Randell (Editors), pp. 88-89. (1976).

[Michel et al, 05]

C.MICHEL, Y.PRIE, L.LE GRAET, Construction d'une base de connaissances pour l'évaluation de l'usage d'un environnement STIC. 17^{eme} Conférence Internationale Francophone sur l'Interaction Homme –Machine, Toulouse, France, p. 199-202. (2005).

[Mille, 06]

A. MILLE, Raisonner à Partir de l'Expérience Tracée. Le storytelling : concepts, outils et applications». Eddie Soulier (dir), *Traité IC2, Série Informatique et SI*, Hermes. (2006).

[Morse et Steves, 00]

E.MORSE, M. STEVES, CollabLogger: A Tool for Visualizing Groups at Work. *Proceedings of WETICE2000, Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises* (Gaithersburg, MD, June 2000), IEEE Computer Society, p.104-109. (2000).

[Najjar et al, 04]

NAJJAR, TERNIERS et DUVALE, User Behavior in Learning Object Repositories: An Empirical Analysis. Educational Multimedia, Hypermedia \& Telecommunications-EDMEDIA04, Lugano, Suisse, p.4373-4379. (2004).

[Nicaud, 87]

J.NICAUD, APLUSIX : un système expert en résolution pédagogique d'exercices d'algèbre, PhD thesis, Thèse de l'Université Paris XI-Orsay. (1987).

[Ollagnier, 03]

M. OLLAGNIER-BELDAME, Comment intégrer la réutilisation des expériences d'apprentissage en FOAD ? Le projet EPICéA. Colloque Environnements Informatiques pour l'Apprentissage Humain. (annexes).(2003).

[Ollagnier, 07]

M. OLLAGNIER-BELDAME, Traces d'interactions et processus cognitifs en activité conjointe : Le cas d'une co-rédaction médiée par un artefact numérique. Thèse de doctorat de l'Université Louis Lumière Lyon 2. (2007).

[Pernin, 03]

J-P. PERNIN, A propos d'objets pédagogiques..., *INRP-ERTE E-Praxis, Laboratoire CLIPS-IMAG. (NOV 2003).*

[Pernin, 05]

J-P.PERNIN, CSE, un modèle de traitement de traces. Rapport interne de recherche CLIPS-IMAG. (2005).

[Pernin et Lejeune, 05]

J-P.Pernin, A.Lejeune, Modèles pour la réutilisation de scénarios d'apprentissage, colloque TICE 2005, Compiègne. (2005).

[Plaisant et al, 96]

C.PLAISANT, B.MILASH, A.ROSE, S.WIDO et B. SHNEIDERMAN, Lifelines: visualizing personal histories. CHI'96: Proceedings of the SIGCHI conference on Human factors in computing systems. ACM Press, New York, NY, USA, p. 221–227. (1996).

[Plaisant et al, 99]

C.PLAISANT, A.ROSE, G.RUBLOFF, R.SALTER et B.SHNEIDERMAN, The Design of History Mechanism and Their Use in Collaborative Educational Simulations. *Proceedings of the Computer Support for Collaborative Learning*. Palo Alto, CA, p. 348-359. (1999).

[Rebai et Labat, 04]

I.REBAÏ, J-M. LABAT, Des métadonnées pour la description des composants logiciels pédagogiques, In Technologies de l'Information et de la Communication dans les Enseignements d'Ingénieurs et dans l'Industrie, pp. 80--87. (2004).

[Rekik, 99]

Y.A. REKIK , C. VANOIRBEEK, Course designer environment: Authoring environment for interactive courses with a shared course component library. In CAEE'99, pages 296–3030. (1999).

[Sanderson, 94]

P. M.SANDERSON, J.SCOTT, T.JOHNSTON, J.MAINZER, M.WATANABE L et J.M. James, MacSHAPA and the enterprise of Exploratory Sequential Data Analysis (ESDA). International Journal of Human-Computer Studies, 41, 633-68. (1994).

[Séjourné et al, 04]

A.SÉJOURNÉ, M.BAKER, K.LUND et G. MOLINARI, Schématisation argumentative et co-élaboration de connaissances: le cas des interactions médiatisées par ordinateur. Actes du colloque international "Faut-il parler pour apprendre?". E. A. Théodile Lille 3 (Ed.), Arras, France. (2004).

[SEMMak, 98]

F.SEMMAK, Réutilisation de composants de domaine dans la conception des systèmes d'information, Thèse de doctorat de l'Université Paris I. (FEVR 1998).

[Stefanov et Stefanova, 05]

J.Stefanov, E.Stefanova, Analysis of the Usage of the Virtuoso System. Workshop

“Usage analysis in learning systems” AIED 05, p.109-110. (2005).

[Settouti et al, 06]

L.S.SETTOUTI, Y.PRIE, A.MILLE et J-C.MARTY, Système à base de traces pour l'apprentissage humain. TICE Colloque international en «Technologies de l'Information et de la Communication dans l'Enseignement Supérieur et l'Entreprise », Toulouse. (2006).

[Settouti et al, 07]

L-S.SETTOUTI, Y.PRIE, J-C.MARTY et A.MILLE, Vers des systèmes à base de traces modélisées pour les eiah, STICEF : numéro spécial Analyses des traces d'utilisation dans les EIAH. (2007).

[Settouti, 11]

L.S. SETTOUTI : Système à base de traces Modélisées : Modèles et langage pour l'exploitation des traces d'Interactions. Thèse de doctorat de L'Université Claude Bernard Lyon 1. 14. (JANV 2011).

[StermseK et al, 07]

G. STERMSEK, M.STREMBECK, G. NEUMANN, A User Profile Derivation Approach based on Log-File Analysis. *International Conference on Information and Knowledge Engineering*, Las Vegas, Etas-Unis. (2007).

[Tchounikine, 02a]

P.TCHOUNIKINE, Quelques éléments sur la conception et l'ingénierie des EIAH, in acte du DDR 13, p 233-245. (2002).

[Tchounikine, 02b]

P.TCHOUNIKINE, Pour une ingénierie des Environnements Informatiques pour l'Apprentissage Humain, Tchounikine P., « Pour une ingénierie des Environnements Informatiques pour l'Apprentissage Humain», Revue I3 information – interaction – intelligence 2(1), www.revue-i3.org. (2002).

[Wexelblat et Maes, 99]

A.WEXELBLAT, P. MAES, Footprints: History-rich tools for information foraging. Rapport de recherche bibliographique, (MARS 2005)

Sites web:

[Ariadne, 96]

Fondation ARIADNE (1996). <http://www.ariadne-eu.org> , (consulté le 26/11/2011).

[eLycée, 11]

<http://www.elycee.com>. (Consulté le 26/11/2011)

[FOAD, 05]

FOAD : le guide pratique FOAD, Un guide pratique pour la formation ouverte et à distance. Université de Lorraine, France.2005. Disponible sur : <http://foad.inffolor.org> (consulté le 13/09/2011)

[http, 1]

http://fr.wikipedia.org/wiki/Environnements_informatiques_pour_l%27apprentissage_humain

[http, 2]

<http://hci.ece.upatras.gr/synergo/>

[http, 3]

http://unit-tice.emn.fr/article.php3?id_article=585

[http, 4]

<http://ltsc.ieee.org/wg12/index.html>

[http, 5]

<http://www.graphic-evolution.fr/definition/fichier-log-106.html>

[http, 6]

Un outil commercial développé par Noldus. www.noldus.com

[http, 7]

Un outil Open source pour analyser et transcrire des vidéos, etc. <http://www.transana.org/>

[IA, 05]

Interaction Analysis – Supporting participants in technology based learning activities.
Kalidoscope projet, <http://www.no-kaleidoscope.org>.