

Visualisation interactive de traces et réflexivité : application à l'EIAH collaboratif synchrone eMédiathèque

Damien Cram, Denis Jouvin, Alain Mille

[LIRIS UMR 5205 CNRS, Université de Lyon]

■ **RÉSUMÉ** : Cet article présente un outil de visualisation interactive de traces d'interactions dans le cadre d'une activité d'apprentissage collaboratif synchrone. Cet outil a été développé en collaboration entre l'entreprise eLycée S.A.S., et une équipe de recherche travaillant sur l'ingénierie de l'expérience tracée et les EIAH. L'hypothèse de facilitation de la tâche d'apprentissage par les processus métacognitifs liés à une activité réflexive est à la base de la contribution. L'article est l'occasion de situer précisément les enjeux du travail engagé, de décrire l'environnement et les outils développés, et de présenter les propriétés des modèles sous-jacents. Bien que cet outil de visualisation de traces n'ait pas encore fait l'objet d'expérimentations, les tests techniques auprès d'un public varié ont rencontré une forte adhésion. L'article pointe les aspects génériques des mécanismes de traçage développés, en particulier les possibilités de faire évoluer dynamiquement l'environnement par l'utilisateur mais aussi par les concepteurs d'activités et les enseignants.

■ **MOTS CLÉS** : Traces d'utilisation ou d'interactions, plate-forme collaborative synchrone, apprentissage à distance, visualisation interactive de traces, réflexivité.

■ **ABSTRACT** : This paper presents an interactive visualization tool for interaction traces, in the context of a synchronous collaborative e-learning activity. This tool has been developed by an e-learning company, eLycée, in collaboration with a research team working on e-learning, interaction traces, and experience reuse. This work adopts the grounding hypothesis that metacognitive processes and reflexive activities can facilitate learning. The paper presents the specific issues of such usage trace construction and visualization, the design of the hosting collaborative synchronous platform, and the associated tools and underlying models properties. This tool has so far not been experimented with students, but the technical tests with various researchers have been very promising and encountered a wide acceptance. The paper also points out the generic aspects of the tracing mechanisms, and the possibility for the student, teacher and designer to configure, update and extend it dynamically.

■ **KEYWORDS** : Usage traces, interaction traces, synchronous collaborative platform, e-learning, interactive trace visualization, reflexivity.

- [1. Introduction](#)
- [2. État de l'art](#)
- [3. Plate-forme collaborative synchrone eMédiathèque](#)
- [4. Visualisation interactive de traces](#)
- [5. Exemple et scénario d'application](#)
- [6. Extensibilité et généricité de l'approche](#)
- [7. Conclusion, discussions et perspectives](#)
- [BIBLIOGRAPHIE](#)

1. Introduction

Il existe entre autres deux types d'usage des traces dans les EIAH. Le premier concerne la collecte des traces d'interactions lors d'activités d'apprentissage médiées par un EIAH pour les analyser ensuite. Le deuxième consiste à élaborer des indicateurs à partir de traces d'interactions pour mettre à disposition des apprenants ou des enseignants des tableaux de bord de l'activité en cours. Mais dans ce deuxième cas, les traces d'interactions liées à l'activité sont en général "cachées" à celui qui les produit. Dans cet article nous adoptons l'hypothèse que la visualisation des traces par celui qui les produit favorise les processus métacognitifs lors de l'activité d'apprentissage, notamment les processus liés à la réflexivité. Notre objectif ici est de préciser les propriétés attendues d'un EIAH collaboratif synchrone tracé, et de proposer des principes, des méthodes et les premières pistes pour concevoir de tels environnements. Une deuxième étape de notre démarche de recherche est d'évaluer le degré de réflexivité supplémentaire apportée par la visualisation réflexive de traces chez l'apprenant lors de ses activités. Cette deuxième étape n'est pas l'objet de l'article, elle fera l'objet de recherches futures, plus orientées vers les sciences humaines. Le concept de réflexivité est introduit ici comme principale motivation dans la conception du

système de visualisation de traces que nous présentons dans la suite.

Après avoir précisé le contexte de ces recherches (section 1), nous parcourons l'état de l'art en matière de systèmes à base de traces dans les EIAH et de visualisation de traces (section 2), et nous positionnons ce travail par rapport au CSCW et au CSCL. Deux points principaux sont ensuite abordés dans l'article : le premier est la présentation de la plate-forme d'apprentissage en ligne *eMédiathèque* (section 3), et de son architecture à base de réification d'actions qui permet un traçage précis lors des activités ; le deuxième est l'explication du fonctionnement du système de visualisation de traces que nous proposons d'installer dans *eMédiathèque* (section 4). La section 5 présente l'interface de l'outil de visualisation et l'illustre par un scénario de collaboration très simple : le cas d'une activité de co-translation de bandes dessinées par un binôme d'étudiants. La restriction de l'activité à un binôme plutôt qu'à un groupe appauvrit certes la dimension collaborative de l'activité mais a pour avantage de valider l'outil et de présenter son interface de manière plus simple. La section 6 explique comment le traçage d'une activité pédagogique supplémentaire peut être envisagé *a posteriori* grâce à la généralité et l'extensibilité du système. La section 7 présente les perspectives de recherche maintenant ouvertes par la présence d'une telle visualisation interactive de trace selon trois axes : l'apport réflexif du système, les possibilités de partage et de réutilisation d'expérience, et enfin l'*awareness* collaboratif apporté par la présence des traces d'une activité collective dans l'environnement.

1.1. Contexte : eLycée et sa plate-forme eMédiathèque

eLycée S.A.S. est une jeune entreprise innovante proposant un service d'enseignement à distance du français, adressé en premier lieu à des étudiants français ou francophones résidant à l'étranger, et n'ayant pas la possibilité, pour des raisons financières ou géographiques, d'intégrer un lycée ou une école française dans le pays d'accueil. Pour ce faire, eLycée¹ a développé un environnement de classes virtuelles, fondé sur des technologies récentes en matière de vidéoconférence et de partage collaboratif. A la différence de nombreux services d'enseignement à distance plus classiques, le modèle de classes virtuelles d'eLycée est un modèle synchrone, dans lequel les élèves et l'enseignant se retrouvent simultanément, communiquent et collaborent dans un espace virtuel par le biais d'outils collaboratifs synchrones et multimédias, dans un contexte technique de vidéoconférence².

Les principales caractéristiques du modèle pédagogique d'eLycée sont d'une part d'instaurer l'élève comme principal acteur du processus d'apprentissage, et d'autre part de favoriser les processus métacognitifs liés à la réflexivité. Pour y parvenir, eLycée base les activités pédagogiques sur la co-construction de documents multimédia (textuels, vidéo, audio ou mixtes), en binômes ou en groupes. Les documents ainsi produits peuvent ensuite être sélectionnés et exposés aux parents ou professeurs par le biais d'un portfolio, actuellement sous la forme d'un blog de classe ou d'élève.

La facilitation des processus métacognitifs est, quant à elle, liée aux outils collaboratifs et au traçage, de façon à ce que les élèves puissent avoir une vision de l'activité des autres, de l'activité de leur groupe, et de leur propre activité. Des phases ou exercices mettant en œuvre explicitement cette réflexivité peuvent ainsi être incluses dans l'activité même par les enseignants : critiques mutuelles lors de l'édition d'un tableau blanc, correction des productions des autres, revue des erreurs d'écriture commises, des mots ou tournures ayant posé des difficultés, etc.

Étant donnés ces besoins originaux, innovants par rapport à l'offre des outils habituels, eLycée a développé *eMédiathèque*. *eMédiathèque* est une plate-forme collaborative synchrone incluant une infrastructure de traçage, un système de visualisation interactive de la trace, et un certain nombre d'outils collaboratifs tracés.

1.2. eMédiathèque, une plate-forme de CSCL tracée

Une tendance que l'on peut constater aujourd'hui dans l'enseignement à distance est celle du *constructionnisme* (Papert, 1993), et plus particulièrement du *constructionnisme social*, dans laquelle les apprenants sont impliqués dans la création de documents ou d'artefacts, de façon collaborative, afin d'améliorer leur apprentissage et leur ressenti. La démarche pédagogique d'eLycée (eLycéeMethod, 2007) va tout à fait dans le sens de cette approche.

Il nous paraît important de souligner que la synchronie des interactions, rendue possible par les plates-formes dites *collaboratives synchrones*, donnent à l'enseignement à distance en classes virtuelles une véritable dimension collaborative, vivante, et vécue comme telle par les apprenants et l'enseignant. Elle permet un degré de collaboration beaucoup plus conscient que le mode asynchrone. Le bouclage action – *feedback* entre les participants étant beaucoup plus rapide, les interactions sont plus efficaces, et l'*awareness* de groupe en est également renforcée.

L'utilisation d'un environnement informatique pour effectuer une activité médiée d'apprentissage collective contraint évidemment les modalités de communication, mais elle présente par contre une propriété unique impossible dans les conditions du présentiel ou de médiation par des outils de simple communication (téléphone, visioconférence simple, etc.) : les interactions entre acteurs et entre acteurs et environnement, passent obligatoirement par l'environnement informatique, elles peuvent donc être représentées sous une forme codifiée, et éventuellement "réifiées" lorsque cette codification est associée à une sémantique d'un niveau suffisant, et peuvent donc être mémorisées, et exploitées en tant que ressources nouvelles pour l'activité. L'environnement informatique particulier qu'est la plate-forme collaborative apporte donc, outre des facilités de co-construction de documents, une *mémoire potentielle des interactions*. Notre travail sur les traces vise à exploiter au mieux cette mémoire.

1.3. Visualisation interactive de traces en temps réel

Nous revendiquons donc le statut de ressources complémentaires pour les traces d'interactions et il convient que ces ressources soient disponibles au moment même de l'activité, avec les outils adaptés à leur utilisation métacognitive par les acteurs de la situation d'apprentissage. La particularité d'eMédiathèque est de permettre le traçage en temps réel, ce qui ouvre un certain nombre de possibilités, notamment dans l'assistance à l'apprenant. Ce que nous présentons ici est un système de visualisation interactive par l'apprenant de ses propres traces d'utilisation, pendant le cours de son activité. L'objectif à la base de notre démarche est ambitieux : offrir un terrain supplémentaire et spécialisé pour faciliter la réflexivité de l'apprenant dans ses activités pédagogiques. Nous désignons ici par activité réflexive "une activité se prenant, au moins en partie, elle-même pour objet" (Ollagnier-Beldame, 2006). L'intérêt apporté par la *réflexivité* au cours de l'activité a été pointé dans de nombreux travaux (Mollo, 2002), notamment lorsqu'elle met en œuvre des processus d'apprentissage (Masciotra, 2005).

Pour que l'apprenant puisse reconnaître son activité dans sa trace, il faut qu'elle soit signifiante dans le cadre de son activité, et chaque fois que l'apprenant effectue une action significative, celle-ci doit apparaître dans sa trace. Si la trace est trop verbeuse ou trop silencieuse, l'apprenant risque de ne pas y prêter attention et ne sera pas encouragé à consulter son expérience d'utilisation. Le système de collecte d'eMédiathèque fournit une trace modélisée initiale, qui décrit les interactions dans un langage orienté outil et exhaustif ; nous l'appelons *trace première*. Pour créer une trace significative et de grain judicieux à partir de cette trace, nous mettons en place un mécanisme de transformation en deux étapes. Premièrement, une *transformation statique* permet d'interpréter et d'abstraire la trace première dans le contexte de l'activité réalisée. Deuxièmement, un ensemble de *transformations dynamiques* permet à l'utilisateur de manipuler sa trace décrite dans le langage de l'activité, pour y faire apparaître les éléments qu'il souhaite.

2. État de l'art

2.1. Plates-formes collaboratives synchrones et de vidéoconférence orientées enseignement

Nous désignons par plate-forme collaborative synchrone, un environnement et ensemble d'outils collaboratifs permettant à n participants à une session collaborative de communiquer via différents médias, comprenant généralement les messages instantanés (le *chat*) et si possible l'audio et la vidéo, et via le partage, et donc la co-construction, de documents électroniques. Le terme synchrone signifie que toute communication ou partage s'effectue en temps réel, par opposition au mode asynchrone ou différé. Le partage d'un document, tel qu'un tableau blanc, consiste à rendre collectives toutes les modifications faites sur le document (à une granularité donnée), en temps réel. Notons que la messagerie instantanée peut être vue comme un document partagé dont la seule modification possible est l'ajout de messages.

Le marché de la formation à distance, que ce soit dans le milieu éducatif ou en entreprise, est en pleine croissance, et il existe de nombreuses plates-formes collaboratives synchrones, avec vidéoconférence, comprenant des outils collaboratifs tels que messages instantanés, tableau blanc partagé, co-navigation et partage d'application.

Parmi les plus complètes, nous pouvons notamment citer la plate-forme *Illuminate*², la plate-forme *Wimba*⁴, la plate-forme *Breeze*⁵ d'Adobe, la plate-forme *Marratech*⁶, ou encore la plate-forme *LearnLinc* de *iLinc*⁷. Ces plates-formes de vidéoconférence sont souvent associées dans leur exploitation à des systèmes de gestion de contenu pédagogique tels que *Blackboard*⁸ ou *Moodle*⁹. Elles apportent avant tout une dimension *synchrone* et *collaborative* à l'enseignement à distance.

Il est intéressant de constater que ces plates-formes regroupent des fonctionnalités grandement similaires, même si elles varient en qualité, à savoir :

- la vidéoconférence, $n-n$ pour l'audio, souvent limitée à $1-n$ pour la vidéo ;
- la messagerie instantanée, avec émoticônes, collective et parfois privée ;
- une liste active des participants, indiquant leur statut, et permettant des interactions simples de type "lever la main" ou "répondre par oui ou par non" ;
- la co-navigation Web, généralement sur une seule page à la fois, ou encore la projection partagée de présentations (de type PowerPoint), page par page ;
- le partage d'application (de type VNC), où les participants voient le bureau ou une partie du bureau, par exemple la fenêtre d'une application, d'un des participants ;
- et, le plus souvent, l'enregistrement des sessions.

Cette dernière possibilité nous intéressera particulièrement : elle peut s'apparenter à une forme rudimentaire de traçage. Toutefois, telle une séquence vidéo, la trace résultante est totalement monolithique et opaque, et ne peut qu'être rejouée de bout en bout : elle reste donc peu exploitable dans notre contexte et notre problématique. Un tel enregistrement ne peut en effet pas être représenté de manière discrétisée et graphique simple ; la présenter à

l'utilisateur en temps réel n'a que peu de sens tant il sera difficile de "s'y retrouver" rapidement.

Étant des logiciels propriétaires, ces plates-formes sont d'autre part très difficiles voire impossibles à tracer (c'est-à-dire à instrumenter en vue de traçage). Malheureusement, aucune solution Open Source n'atteint actuellement le niveau de finition, de complétude et de performance requis pour ce type d'activité. Enfin, ces plates-formes intègrent généralement mal le contenu de type *rich media* dans leurs différents outils collaboratifs.

2.2. Conception et traitement informatique de la trace dans les EIAH

Nous appelons "trace d'interactions" tout objet informatique dans lequel s'accumulent des données à propos des interactions entre un système informatique et son utilisateur. Dans les EIAH, comme dans les autres environnements informatiques, les traces d'interactions constituent donc une source d'information sur l'expérience d'utilisation et font l'objet de nombreux travaux de recherches qui tentent de mettre à profit cette source d'expérience. (Hilbert et Redmiles, 2000) décrivent des techniques pour extraire des connaissances sur la facilité d'utilisation des systèmes informatiques à partir des événements de l'interface. Des outils d'analyse des comportements à partir de données séquentielles sont également étudiés, et formalisés par (Sanderson et Fisher, 1994).

Mais l'idée de réutiliser informatiquement l'expérience pose le problème fondamental de la modélisation de celle-ci. (Iksal et Choquet, 2005) utilisent le langage UTL (*Usage Tracking Language*) pour représenter le plus génériquement possible la trace d'une activité pédagogique. L'approche MUNETTE (*Modéliser les Usages et les Tâches pour Tracer l'Expérience*) ne se limite pas aux EIAH et propose un cadre générique pour la modélisation de l'expérience dans le cadre de sa réutilisation pour l'assistance à base de traces. Une implémentation de Musette a été réalisée dans le cadre de la conception assistée par ordinateur (Champin, 2003). D'autres systèmes utilisent les traces d'interactions à des fins d'analyse *a posteriori* des usages et des activités, soit dans le but de quantifier certains usages (Rossi et al., 2005), soit pour extraire des motifs fréquents dans l'activité tracée (Georgeon et al., 2006) et étudier les comportements des utilisateurs à partir de ces motifs.

On peut trouver des réutilisations des traces dans les EIAH aussi bien pour l'assistance à l'apprenant que pour l'analyse de son activité. Mais quel que soit le type de réutilisation des traces d'interactions qui est faite, le besoin de définir un cadre formel commun à tout travail sur les traces se fait sentir. Nous le résumons ci-dessous en nous fondant sur les travaux développés dans le cadre du Projet "Personnalisation des EIAH" du Cluster ISLE de la région Rhône-Alpes¹⁰.

2.3. Le Système à Base de Traces modélisées

2.3.1. Architecture du Système à Base de Traces modélisées

Le *Système à Base de Traces modélisées* (SBT) (Settouti et al., 2007) formalise les traitements à appliquer à la donnée "trace" depuis sa collecte jusqu'à ses applications dans un cadre très générique. Il s'applique à tout type d'environnement tracé, en particulier aux EIAH. La gestion des traces dans eMédiathèque respecte ce type d'architecture. Comme le montre la [figure 1](#), un SBT est composé de plusieurs modules interdépendants.

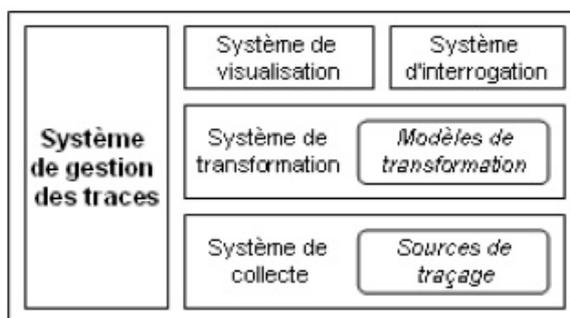


Figure 1. Schéma simplifié d'un Système à Base de Traces

Le *système de collecte* capture les interactions par l'intermédiaire de *sources de traçage*, et crée une première trace : la "trace première". Le *système de transformation* constitue le cœur du SBT. Il permet d'obtenir d'autres traces à partir des traces collectées. Le choix du *modèle de transformation* à appliquer à une trace dépend de l'intention d'utilisation de cette trace. L'ensemble des traces collectées et transformées est alors accessible par l'intermédiaire d'un *système de requête* et d'un *système de visualisation*. Ce *système de visualisation* doit donner la possibilité d'avoir une vue ergonomique et potentiellement interactive sur les traces du SBT, afin de permettre l'analyse et l'interprétation de celles-ci. Le système de visualisation peut s'appuyer sur le *système de requête*. A l'heure actuelle, les systèmes utilisant des traces possèdent généralement un module de visualisation, mais le fonctionnement d'un *système de visualisation* générique n'a pas été étudié. L'exploration d'un système générique de visualisation interactive de trace est abordée en section 3.

2.3.2. Enjeu de la collecte de la trace première

L'étape de collecte de la trace première est cruciale, car la valeur de tout travail reposant sur des formes transformées ou dérivées de celle-ci dépend de sa qualité. Par "qualité", nous entendons surtout son exhaustivité dans les objets d'intérêt tracés et son grain. Il est souvent argumenté que les fichiers *logs* constituent un bon support d'analyse de l'expérience d'utilisation d'un système (Roussel et al., 2006). En pratique, réutiliser les *logs* à ces fins mène à se confronter à des problématiques de manque de contexte des informations qu'ils véhiculent, de manque de structure dans ces données, de trop grand volume de données à traiter par rapport au besoin, ou encore d'inadéquation des données pour l'analyse d'une activité.

C'est pourquoi toute analyse à base de *logs* doit s'accompagner d'un effort de restructuration et de remise en contexte des informations dans l'activité, par exemple par le biais de la visualisation d'une vidéo de l'activité loguée (Avouris et al., 2005). De plus, les fichiers *logs* sont bien souvent des données accessibles depuis le serveur, ce qui ne permet pas de les réutiliser en local et en temps réel comme nous le souhaitons dans le cas de la visualisation interactive de traces pour la réflexivité.

Notons qu'un fichier log est un objet créé par le concepteur pour être utilisé à des fins d'information technique, pour le débogage ou le dépannage par exemple. C'est la raison pour laquelle il est inapproprié de le réutiliser pour des analyses de haut niveau sémantique comme l'interprétation d'une activité.

Les traces, au sens où nous l'entendons dans cet article, sont des objets spécifiquement réalisés pour la modélisation et la réutilisation de l'expérience d'un utilisateur. La trace est le plus souvent *trace d'interactions*, c'est-à-dire que la donnée source qui permet de la construire est l'interaction humain / machine. La trace s'enrichit à chaque nouvelle interaction au fur et à mesure que l'activité se déroule, en prenant en compte le contexte de chaque action réalisée. C'est donc un objet dynamique, qui peut être disponible en temps réel et qui peut être modifié, requêté ou visualisé.

2.4. Visualisation de traces et de données s'apparentant à des traces

Dans (Ollagnier-Beldame, 2006), l'auteur distingue entre autres deux catégories de systèmes utilisant 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.

Dans la première catégorie, on trouve la représentation *LifeLine* de (Plaisant et al., 1996) qui permet facilement au lecteur de remettre en contexte des éléments ponctuels d'une histoire ou de corréler certains événements grâce à des lignes de vie superposées sur un axe temporel linéaire horizontal. 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., 1999). Avec *PadPrints* (Hightower et al., 1998), une carte de navigation représente cette histoire. Parfois, c'est l'expérience d'utilisation de plusieurs utilisateurs qui est visualisée. Le système *FootPrints* (Wexelblat et Maes, 1999) permet aux utilisateurs de laisser des annotations en format texte sur les éléments significatifs de l'environnement pour les utilisateurs futurs. La visualisation des traces d'une activité pédagogique proposée par (Marty et al., 2007) permet de mettre en avant des périodes significatives de l'activité où l'apprenant est en difficulté grâce à des zones d'ombre.

Dans le cas de l'EIAH *PIXED* (Heraud et al., 2004), 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.

D'autres systèmes réutilisent les traces qui proviennent de sessions d'apprentissage pour établir une vue d'ensemble de la classe. (France et al., 2007) utilisent les faces de Chernhoff (Chernhoff, 1973) pour représenter en temps réel l'état d'avancement de chaque élève dans son activité d'apprentissage. Les faces de Chernhoff de tous les élèves sont disposées dans des graphiques permettant de visualiser l'état d'avancement général d'une classe par rapport à l'état d'avancement de chaque individu, comme pourrait le faire un enseignant en présentiel. Le système *CourseVis* de (Mazza et Dimitrova, 2003) est comparable, mais utilise des matrices (2D ou 3D) et des sphères de tailles différentes pour représenter les scores de chaque étudiant pour les activités qu'il a effectuées selon plusieurs dimensions.

Enfin, les études faites avec l'environnement DREW (*Dialogical Reasoning Educational Web tool*) (Séjourné et al., 2004) 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.

2.5. Synthèse : les particularités liées au "collaboratif temps réel"

Il existe des théories et des classifications générales de la visualisation de données, et notamment sur le sens que la

structure d'une représentation est capable de transmettre à son lecteur (Lohse et al., 1994), (Novick et Hurley, 2001). Cependant il n'est pas facile d'en déduire comment construire une visualisation de traces d'interactions personnelles pour la réflexivité. En pratique, ce choix dépend du contexte d'application des traces. Ce qu'il y a de commun à toutes les visualisations de données s'apparentant à des traces, c'est la composante temporelle et historique forte à faire valoir lors de la représentation. Les expérimentations menées sur les systèmes présentés en section précédente montrent globalement que les utilisateurs sont disposés à faire l'effort de consulter leur expérience d'utilisation pour tenter de la mettre à profit dans leur propre activité.

L'approche que nous adoptons dans cet article reprend les conclusions de (Séjourné et al., 2004), mais nous souhaitons que la trace d'activité soit expressément affichée à l'utilisateur pour qu'il existe une conscience de trace chez l'apprenant. L'analyse de l'activité faite à l'aide de l'instrumentation de DREW montre bien que la visualisation et la manipulation par le sujet des traces "spontanées" de son activité sont un moyen d'encourager les processus métacognitifs qui en permettent l'appropriation.

Deux nouvelles contraintes sont apportées par le caractère "temps réel" de la visualisation. Premièrement, la visualisation ne doit pas créer de surcharge cognitive chez l'apprenant, car l'exploitation de ses traces ne constitue pas son activité principale, mais enrichit simplement les possibilités offertes. Pour cela, les fonctions de la visualisation doivent être simples et intuitives ; les opérations possibles sur la trace doivent être faciles d'appropriation. La deuxième contrainte est plus matérielle : l'espace de représentation de la trace est restreint, car l'environnement doit consacrer l'essentiel de la surface de l'écran à l'activité pédagogique. En conséquence, les possibilités en termes d'espaces graphiques sont assez réduites.

Dans eMédiathèque, nous avons choisi de mettre en place des fonctions simples pour l'utilisateur : trier des éléments contenus dans la trace, filtrer certains éléments de la trace, regrouper des éléments, remettre un élément de la trace dans son contexte d'utilisation, segmenter la trace en plusieurs segments ou épisodes, afficher simultanément deux traces différentes pour les comparer, supprimer certains éléments, choisir l'apparence visuelle de certains éléments, etc. D'autres fonctions choisies par un enseignant par exemple pourraient compléter la palette disponible selon une intention pédagogique ou didactique donnée.

Il est clair qu'un enjeu majeur du CSCW, et par voie de conséquence du CSCL, est la notion d'*awareness* collaboratif, comme le montrent (Dourish et Bellotti, 1992), (Gutwin et al., 1996b). Le lien avec les traces collectives tient d'une part du fait que nous traçons une activité collaborative, et que cette trace doit être liée à la conscience des autres pour faire sens, et que d'autre part la visualisation d'une trace collective constitue en soi un outil de collaboration, et promeut, nous l'espérons, l'*awareness* collaboratif (par exemple en permettant de voir l'activité d'autres membres du groupe, même si le résultat de leur action n'est pas visible dans ce qui est partagé). Bien que le deuxième point soit une motivation majeure dans les perspectives de notre travail (cf. perspectives en 7.1 et 7.3), cet article se focalise sur le premier, à savoir les apports et les contraintes de l'aspect collaboratif de l'activité dans la construction et la visualisation de la trace.

3. Plate-forme collaborative synchrone eMédiathèque

Nous désignons par "environnement eLycée" l'ensemble des logiciels et des services en ligne mis à disposition des étudiants et enseignants par eLycée. Cet EIAH adhère au concept de classe virtuelle. Il peut être décrit plus généralement comme une plate-forme collaborative synchrone de *e-learning*, mettant en œuvre différents outils collaboratifs à vocation principalement pédagogique.

L'environnement eLycée est composé principalement des éléments suivants :

- un gestionnaire de contenu pédagogique (CMS pour *Course Management System*), fournissant essentiellement du contenu Web et *rich media* (clips audio ou vidéo), et quelques services web exploités par eMédiathèque ;
- un certain nombre d'outils ou services en ligne intégrés ou référencés dans les pages Web délivrées par le CMS (par exemple l'éditeur de texte collaboratif *writely* de Google, un système de blog, des dictionnaires en ligne, etc.) ;
- et enfin la plate-forme collaborative eMédiathèque.

eMédiathèque est le logiciel client, au sens informatique du terme, développé par eLycée, et constitue le point d'accès à l'environnement eLycée. eLycée fait également usage de la plate-forme Marratech, en complément d'eMédiathèque, pour gérer la vidéoconférence. eMédiathèque intègre Marratech de manière à présenter l'ensemble comme un seul et même environnement à l'utilisateur.

3.1. Fonctionnalités transverses

Au sein de l'environnement eLycée, eMédiathèque a principalement deux rôles : d'une part, un rôle fort d'intégration. eMédiathèque gère à la fois la connexion aux différents services de l'environnement eLycée, par un

mécanisme commun et unique d'authentification (*single sign on*), et l'intégration graphique et ergonomique des différents outils et services dans un tout cohérent ; et d'autre part, la mise en œuvre effective de divers outils collaboratifs synchrones originaux, nécessaires aux activités pédagogiques d'eLycée.

En plus de ces deux fonctions premières, eMédiathèque fournit plusieurs fonctionnalités transverses liées aux besoins d'eLycée évoqués en introduction :

- la mise à jour automatique du logiciel, et en particulier le déploiement automatique de nouvelles fonctionnalités sous forme de *plug-ins* ;
- la prise en compte du *rich media* dans les outils collaboratifs (tableau blanc avec *rich media* ; partage et visualisation synchrones de clips vidéo ou audio, etc.) ;
- une interface utilisateur conviviale et très versatile, permettant l'intégration visuelle de divers outils hétérogènes, entièrement configurable par l'utilisateur ;
- et enfin un système de traçage de l'activité de l'utilisateur, permettant de collecter en temps réel ses interactions avec l'outil, sous forme d'actions cohérentes et contextualisées, et constituant ce que nous nommons la trace première.

3.2. Choix technologiques et architecture réseau

Le choix technologique pour eMédiathèque s'est porté sur l'environnement d'exécution Eclipse RCP pour de nombreuses raisons, dont voici les principales. Il s'agit d'un environnement d'exécution basé sur Java et indépendant du système d'exploitation. Il est compatible OSGi et par là-même fournit les mécanismes de base permettant la mise à jour automatique et le déploiement dynamique de *plug-in*. Eclipse RCP fait l'objet de nombreuses extensions et bibliothèques (nous utilisons notamment *Eclipse Modeling Framework*, voir section [3.5.3](#)), et est doté d'une communauté d'utilisateurs importante.

Du point de vue de l'IHM, l'interface utilisateur proposée par Eclipse RCP est tout à fait adaptée (gestion simultanée de multiples documents, vues périphériques), et paramétrable par l'utilisateur. Les possibilités notamment de réorganisation de l'espace de travail (juxtaposition et empilement d'éditeurs et de vues) sont assez uniques. En outre, les bibliothèques graphiques SWT, et GEF pour le tableau blanc, apportent un niveau de finition et de rendu particulièrement intéressant. Enfin, SWT permet d'intégrer le navigateur Web "natif" du système d'exploitation sous-jacent (Internet Explorer sous Windows, Safari sous Mac OS X, Firefox sous Linux) dans l'application, sous forme d'onglets, or la volonté d'eLycée était initialement de concevoir eMédiathèque comme un navigateur Web enrichi et instrumenté.

Du point de vue de l'architecture réseau, eLycée a choisi de se baser sur le protocole XMPP (*eXtensible Messaging and Presence Protocol*), ainsi que deux extensions répandues (XEP-0045 *group chat*, et Jingle). Ce protocole, recommandé par l'IETF, est suffisamment ouvert et extensible pour répondre aux besoins de partage d'actions d'eMédiathèque : utilisant XML comme syntaxe, il s'harmonise d'autre part bien avec EMF. Il s'agit d'un modèle de communication pair à pair (point à point) dans sa forme la plus simple, mais est tout à fait comparable aux systèmes orientés messages par souscription publication (de type JMS *publish/subscribe*) lorsque l'on considère l'extension *group chat*, et la possibilité d'interconnecter plusieurs serveurs XMPP. Par rapport à des standards de systèmes orientés messages plus généraliste, XMPP présente une sémantique plus proche de nos besoins (mettant en jeu des utilisateurs, des sessions multimédia, des salons de discussion, etc.), et garantit de meilleures possibilités d'interopérabilité avec des outils collaboratifs existants.

3.3. Interface utilisateur configurable

L'interface utilisateur d'eMédiathèque (voir la copie d'écran, [figure 9](#)) est basée sur les concepts de vues et d'éditeurs venant du *framework* Eclipse RCP (*Eclipse Rich Client Platform*). Dans cet environnement, l'utilisateur dispose d'une zone d'éditeurs, encadrée sur la [figure 9](#), dans laquelle chaque éditeur représente un contexte ou un document donné (par exemple un tableau blanc), matérialisé sous la forme d'un onglet. Comme nous pouvons le constater sur la figure, les éditeurs peuvent également être juxtaposés verticalement ou horizontalement, subdivisant ainsi à volonté la zone d'éditeurs en sous zones.

Cette zone d'éditeur est elle-même entourée des vues périphériques, qui peuvent elles aussi être disposées, empilées, ou juxtaposées les unes par rapport aux autres, selon la volonté de l'utilisateur. A la différence des éditeurs, les instances de vues sont généralement uniques : nous trouverons ainsi une seule vue "*Media Files*", etc.

Les principaux intérêts de cette organisation sont : d'une part une facilité accrue d'intégration visuelle et ergonomique de différents outils collaboratifs hétérogènes, présentés sous la forme de vues ou éditeurs autonomes ; d'autre part une très grande *configurabilité* de l'environnement par l'utilisateur ; et enfin une similarité avec les navigateurs Web modernes à onglets tels que Firefox ou Internet Explorer 7.0. En l'occurrence, eMédiathèque est souvent présenté aux utilisateurs comme un navigateur Web enrichi de capacités de collaboration et de traçage.

3.4. Outils collaboratifs d'eMédiathèque

eMédiathèque fournit actuellement quatre principaux outils collaboratifs.

La **navigation Web partagée** (co-navigation) permet de naviguer sur des pages Web de manière collaborative, en effectuant des *push* Web (envoi d'une page vers les autres participants de la classe). Les enseignants peuvent alors utiliser cet outil comme mécanisme de présentation collective. Les plates-formes de vidéoconférence citées dans l'état de l'art fournissent généralement cette fonction, toutefois elles ne permettent l'affichage que d'une seule page à la fois, et le type de contenu Web pouvant être affiché est souvent très limité. eMédiathèque autorise l'ouverture de plusieurs onglets de navigation partagés simultanément, ce qui permet non seulement d'organiser le cours de façon plus élaborée (par exemple, mettre en vis-à-vis deux séries de pages, l'une d'un exercice et l'autre de son corrigé), mais aussi facilite l'usage par les élèves de cet outil, car ils peuvent ainsi intervenir sur leurs propres sessions de navigations, en étant autonomes, sans pour autant être intrusifs sur la session de co-navigation de l'enseignant.

La **visualisation synchrone de ressources multimédia** est une fonctionnalité unique d'eMédiathèque qui permet de diffuser dans le cadre d'une classe des fichiers multimédia, de type image, vidéo, audio ou flash, par le biais d'eMédiathèque. Bien qu'à première vue redondante avec l'affichage de contenu Web, elle a deux intérêts majeurs : d'une part, un programme résident (comparable au mode résident ou iconisé des logiciels de messagerie instantanée tels que MSN ou Skype) permet de pré télécharger ces ressources multimédia volumineuses avant la classe, ce qui évite les délais et les problèmes de bande passante ; d'autre part, les ressources ainsi gérées par eMédiathèque peuvent être lues de façon synchrone : un participant à la classe peut "pousser" une ressource particulière (comparable aux *push* Web), puis démarrer, arrêter la lecture, etc., de manière partagée. Les ressources peuvent bien entendu être également lues de façon autonome par les utilisateurs. Un autre avantage pour nous est que la navigation (au sens de lecture) du clip multimédia est tracée.

Le **tableau blanc partagé**. Similaire aux tableaux blancs partagés présents dans des plates-formes telles que Marratech ou Elluminate, il offre toutefois la possibilité d'ouvrir plusieurs tableaux blancs simultanément, et permet donc la juxtaposition de deux tableaux blancs, ou bien d'un tableau blanc avec une page web ou une ressource multimédia. Cette possibilité de juxtaposition facilite la mise en œuvre de nombreux scénarii pédagogiques, tels que les commentaires de clips vidéo, d'images, de chansons, ou autres. Enfin, le tableau blanc d'eMédiathèque est entièrement tracé, et, en résonance avec la visualisation interactive de la trace, est l'objet d'un certain nombre d'applications auxiliaires de la trace telles que : défaire / refaire ; revenir à un état particulier du tableau blanc ; recréer des objets effacés ou hors contexte.

Enfin, la **messagerie instantanée** est comparable aux messageries instantanées présentes dans la plupart des plates-formes collaboratives synchrones, et est tracée.

3.5. Mise en œuvre des traces dans eMédiathèque

3.5.1. Architecture logicielle centrée sur les actions tracées

La plate-forme eMédiathèque a été conçue dès l'origine avec l'idée d'une part de tracer les actions significatives de l'utilisateur, et d'autre part de permettre le partage réseau de ces actions. En conséquence, son architecture s'articule autour des notions d'**action** et de **commande** utilisateur¹¹. Une caractéristique importante des actions est que, même à ce niveau "proche de l'outil", elles doivent être obligatoirement perceptibles, c'est-à-dire donner lieu à un *feedback* visible par l'utilisateur dans l'interface.

Il est important de noter que toutes les traces produites sont collectées par le logiciel client, à la source, et sont totalement inaccessibles au serveur. Notre éthique par rapport à l'aspect « privé » de la trace est qu'une trace diffusée ou partagée doit l'être explicitement. Par défaut, seules les actions déjà partagées, donc publiques pour le groupe, sont accessibles dans la trace collective.

Les actions manipulent un modèle (ou représentent la manipulation d'un modèle), comprenant des entités structurées, elles-mêmes représentant l'état courant des différents documents ouverts, et de leur contenu. Ces actions et entités sont tout à fait compatibles avec les actions et entités observables d'un modèle de trace, et constituent donc dans le cadre d'eMédiathèque le modèle de trace première.

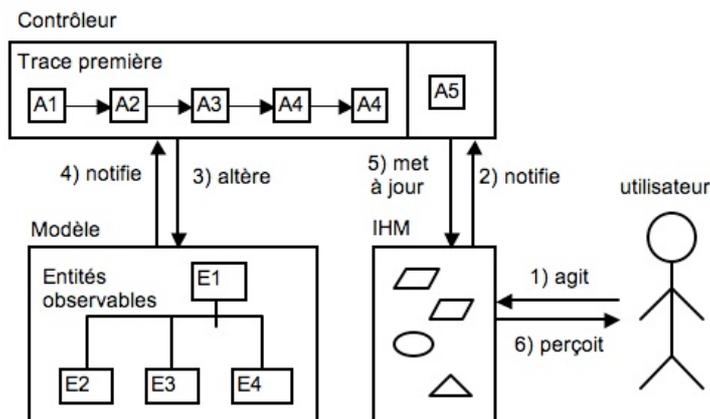


Figure 2. Architecture suivant le *design pattern* MVC centrée sur les actions tracées

La conception d'eMédiathèque suit le patron de conception MVC (Modèle – Vue – Contrôleur), comme le montre la figure 2, avec une séparation nette entre l'IHM qui constitue la vue, les entités observables qui constituent le modèle, et l'infrastructure d'exécution des commandes qui constitue le contrôleur. D'un point de vue technique, l'originalité d'eMédiathèque est que toute action y est réifiée par un objet la représentant explicitement. Ces actions ou commandes sont exécutable, c'est-à-dire que les classes correspondantes encapsulent le code de l'action permettant de modifier le modèle, et symétriquement de représenter cette modification. La mémorisation par le contrôleur des actions effectuées forme la trace première. Bien qu'orientée outil, puisque définie par le concepteur, la trace première ne contient que des actions perceptibles par l'utilisateur, et reste donc assez proche de son langage.

La figure 2 montre notamment que toute interaction utilisateur / système modifiant potentiellement le modèle passe par le contrôleur (2 sur la figure), puis par le modèle (3 et 4 sur la figure), pour revenir par le contrôleur (5), et arriver sur l'IHM (6). Les commandes jouent à la fois le rôle de "requêtes utilisateur", lorsqu'elles n'ont pas encore été exécutées, et le rôle d'actions observées réifiées, lorsqu'elles ont été exécutées et sont mémorisées dans la trace première. Les commandes contiennent toutes les informations nécessaires à leur exécution : elles représentent en quelque sorte le delta entre deux états du modèle, de manière déclarative. A noter que certaines actions ne modifient pas le modèle, tout en étant perceptibles par l'utilisateur.

3.5.2. Lien entre traçage et partage des actions

La réification des actions utilisateur n'a pas pour seule vocation le traçage des actions dans la trace première, mais également le partage collaboratif du modèle. En effet, un moyen élégant de parvenir à du collaboratif synchrone, est de faire en sorte que toute action sur une entité partagée soit sérialisée et propagée par le réseau, et réinterprétée sur les postes des différents participants à la classe.

Il faut remarquer que, à la différence d'approches de "type VNC", où seuls des événements de très bas niveau (clics de souris, touches de clavier), sans sémantique, circulent sur le réseau, ici des actions déjà structurées et associées à une sémantique de "niveau relativement haut" sont communiquées : cela augmente considérablement les possibilités d'interprétation et de raisonnement sur ces actions.

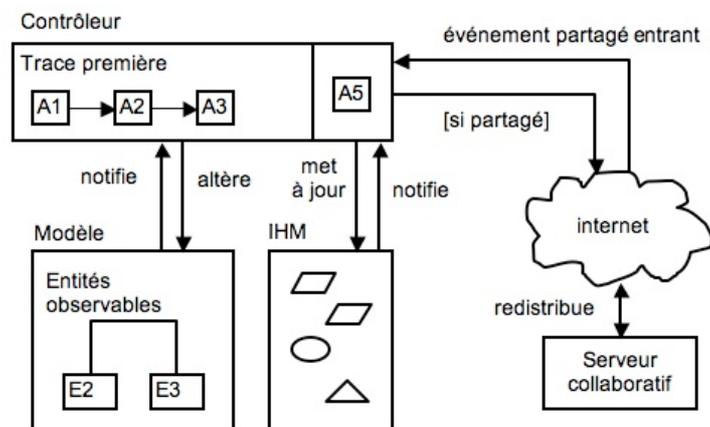


Figure 3. Architecture logicielle centrée sur les actions tracées et partagées

La figure 3 reprend le schéma de la figure 2, en y ajoutant une boucle des actions ou commandes partagées vers le

réseau Internet : le serveur collaboratif a pour fonction de redistribuer les événements, représentés par les actions sérialisées, aux différents participants concernés, et bien sûr de gérer la session globale (connexion et déconnexion) associée à la classe.

Ce principe particulièrement simple de conception permet d'affirmer que dans ce contexte, "tout ce qui est partagé peut être tracé, et tout ce qui est tracé est *a priori* partageable", moyennant relativement peu d'efforts. C'est en partie autour de ce principe qu'a été conçu eMédiathèque.

3.5.3. Utilisation d'EMF pour modéliser la trace

La conception des modèles de trace a été réalisée avec le *framework* EMF d'Eclipse (*Eclipse Modeling Framework*). En quelques mots, ce *framework* permet la génération, l'opérationnalisation et l'exploitation d'un modèle de données structuré, de nombreuses possibilités de réflexivité sur ce modèle, et ouvre des perspectives intéressantes comme par exemple l'édition graphique des modèles de traces au sein même d'eMédiathèque. En outre, EMF facilite la sérialisation en XML/XMI des instances du modèle utilisé dans l'application, à savoir les traces, les documents, les métadonnées liées aux traces, les configurations de visualisation.

Enfin, EMF est de plus en plus utilisé, et fait déjà l'objet d'outils génériques de requêtage (*EMF Query*), de visualisation et d'édition (avec le *Graphical Modeling Framework*), de persistance en bases de données (via *Service Data Object*, ou encore le projet Teneo), ce qui ouvre des perspectives particulièrement enthousiasmantes pour l'analyse de traces.¹²

3.5.4. Exemple de trace première produite

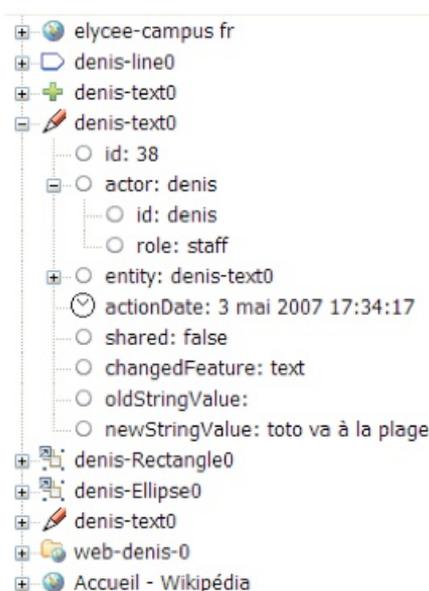


Figure 4. Exemple de trace première produite

La figure 4 présente un exemple d'un très court épisode de trace première produite par eMédiathèque, et visualisé à l'aide d'une vue rudimentaire de la trace première, sans aucune transformation. Cette vue n'est pas destinée à être présentée à l'utilisateur, car trop verbeuse et technique, mais donne un bon aperçu des informations présentes dans la trace première. Les icônes à gauche correspondent aux types d'action, et l'arborescence permet d'accéder à toutes les informations tracées.

4. Visualisation interactive de traces

Dans cette section, nous développons une des possibilités offertes par un système collaboratif tracé en temps réel tel qu'eMédiathèque. Il s'agit de la réutilisation de la trace produite pour l'afficher à l'utilisateur et lui permettre d'interagir avec elle. La visualisation que nous mettons en œuvre dans eMédiathèque est la même pour l'apprenant que pour l'enseignant, c'est pourquoi nous utilisons indifféremment le terme *utilisateur* pour désigner l'individu qui fait usage de la plate-forme et à qui est destiné le système de visualisation de traces. L'enseignant et l'élève disposent donc dans le cas présent des mêmes possibilités de visualisation, en revanche il est fort probable qu'ils n'en feront pas le même usage. Nous discuterons ces points en section 7.

4.1. Définitions

Pour la suite, nous donnons quelques définitions :

- **Trace** : Ensemble structuré qui représente l'histoire d'interaction de l'utilisateur du système et son expérience

d'utilisation. Une définition formelle de la trace est donnée dans (Settoui et al., 2007). Dans cet article on définit une trace comme une séquence d'observés (cf. figure 8).

- Un **observé** est une unité interactionnelle qui compose la trace. Il contient tous les éléments de contexte de l'action effectuée qu'il représente : l'acteur qui la réalise, le type d'action réalisée, la ou les entités impactées, etc.
- Le **modèle de trace** est l'ensemble des concepts et relations types qui décrit les éléments (acteurs, entités et actions) qui peuvent être contenus dans la trace, et comment ils s'organisent dans la trace.
- La **trace première** est la trace directement produite par le système de collecte et décrite dans le langage du concepteur. Toutes les informations sur l'expérience tracée utilisées dans notre système de visualisation proviennent de cette trace première ou bien sont déduites des informations de la trace première.
- Enfin nous reprenons le terme de **trace métier** utilisé par (Georgeon et al., 2006) pour définir une trace qui représente l'expérience d'une activité donnée et qui est décrite dans le contexte de cette activité. Elle contient donc un vocabulaire, et une sémantique compréhensible par l'utilisateur qui verra cette trace.

4.2. Traces centrées utilisateur liées à une activité

Pour parvenir à une trace métier à partir de la trace première, il faut réaliser un certain nombre d'opérations, car la trace première présente deux inconvénients principaux et ne peut être présentée telle quelle à l'utilisateur. Premièrement, elle est trop verbeuse car elle trace toute l'utilisation de la plate-forme de manière exhaustive. Deuxièmement, elle est décrite dans le langage de conception, qui n'est pas facilement compréhensible et inadapté à l'activité que mène l'utilisateur. Nous supposons en effet une plate-forme tracée relativement neutre par rapport à l'activité, dans le sens où elle ne prescrit pas de réaliser une activité plutôt qu'une autre. En conséquence, la trace première contient des éléments d'un niveau sémantique proche de l'outil. Pour obtenir une *trace métier*, il faut que son *modèle de trace* soit plus précis et prenne en compte le cadre et la sémantique de l'activité.

Plus précisément, pour obtenir une trace qui fait sens dans le cadre d'une activité *activité1*, il faut que son modèle de trace (MT_1) décrive le contexte de cette activité, et qu'une transformation *transformation1* construise la *trace métier 1* conforme à MT_1 à partir de la *trace première* conforme au *modèle de trace première* (MT_p). Ainsi, la *trace métier 1* représente l'histoire de l'activité 1 (cf. figure 5). Pour n activités différentes à tracer, il faudra définir n modèles de trace métier et n transformations d'abstraction sémantique. Ces transformations mettent en jeu un certain nombre d'opérations : des suppressions d'observés, des ajouts d'éléments, des substitutions d'observés, etc.

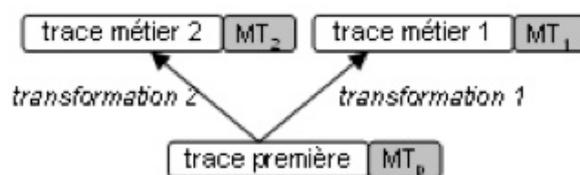


Figure 5. Obtention d'une trace métier

4.3. Transformations de la trace métier par l'utilisateur

La trace métier que nous construisons n'est pas modifiable directement par l'utilisateur. En réalité, pour transformer sa trace métier, l'utilisateur configure un certain nombre de transformations liées à la visualisation. Un *Visualiseur* calcule la *trace vue* par l'utilisateur à partir de la trace métier et de la configuration des transformations (cf. figure 6).

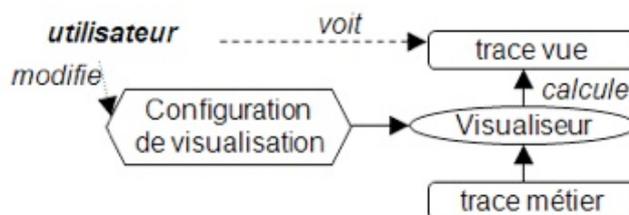


Figure 6. Mécanisme d'interactions avec la trace métier

L'objet *Configuration de visualisation* contient les règles supplémentaires de transformation de la trace métier que le *Visualiseur* applique. On y trouve notamment : une liste des règles de filtrage, une liste des règles de tri, des masques visuels qui définissent quelles propriétés des actions, acteurs et entités sont visibles sur la trace. Chaque configuration représente le "delta" entre la trace métier et la trace vue.

Finalement, la trace vue par l'utilisateur dépend entièrement de la trace métier et de l'objet *Configuration de visualisation*. Si l'un des deux est modifié alors la trace vue est modifiée. Cette approche présente l'avantage de

permettre à l'utilisateur de pouvoir visualiser la même trace métier selon des angles différents, correspondant mieux à la singularité de son activité en contexte.

4.4. Architecture de la visualisation interactive de trace en temps réel

La figure 7 montre l'architecture globale du système de visualisation. A chaque interaction entre l'utilisateur et la plate-forme observée par le système de collecte, une action est ajoutée à la trace première. A chaque activité qui est tracée est attaché un transformateur qui permet de construire la trace métier correspondante à partir de la trace première. Lorsqu'une action est ajoutée à la trace première, tous les transformateurs sont notifiés, ils interprètent la nouvelle action et modifient les traces métier. L'utilisateur de la plate-forme choisit la trace métier qu'il souhaite visualiser (plusieurs traces métier peuvent être visualisées en même temps). A chaque ajout d'action dans la trace métier par le transformateur, le *Visualiseur* recalcule la trace vue, ainsi que lorsque l'utilisateur modifie la configuration de la trace vue.

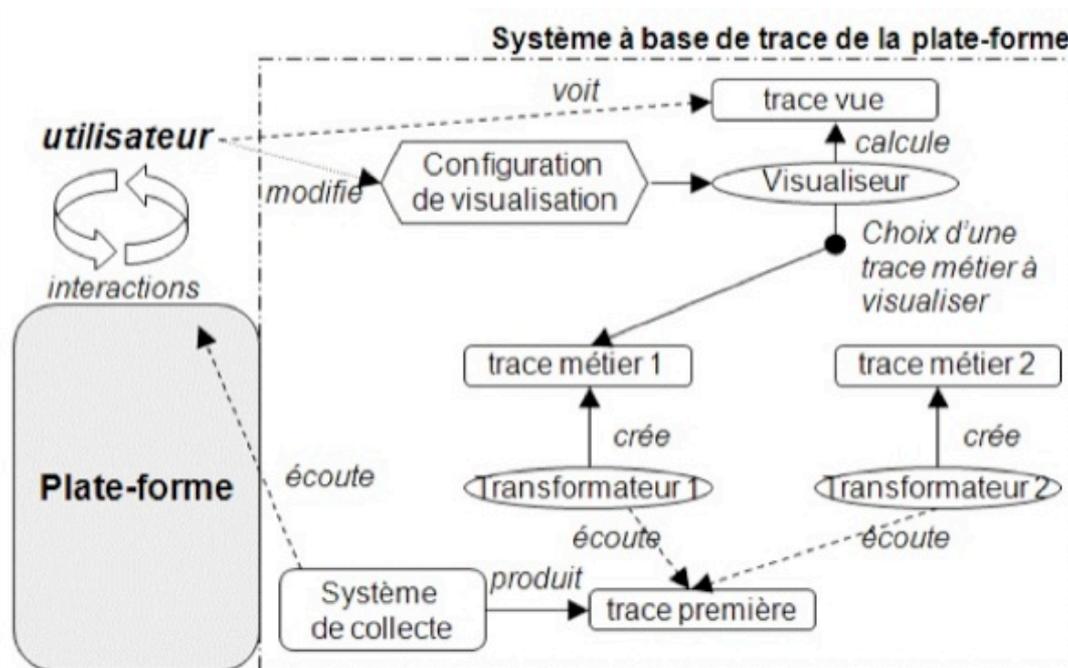


Figure 7. Schéma d'architecture globale de la visualisation interactive de traces

Nous pouvons distinguer deux phases dans le processus de transformation de la trace première en trace vue par l'utilisateur. La première phase consiste à abstraire la sémantique de la trace première pour décrire l'activité à partir des outils et ressources, et non plus l'utilisation de la plate-forme. Nous appelons cette phase **transformation statique**, car elle ne peut pas changer pendant l'exécution. C'est avant le lancement de la plate-forme que les transformations qui permettent d'obtenir les traces métier sont décrites. A ce stade du processus de transformation, la trace est toujours un objet informatique, mais les données qu'elle contient ont été sélectionnées et éventuellement créées pour une activité métier donnée.

La deuxième phase, celle qui permet à l'utilisateur une interaction avec la trace, est appelé **transformation dynamique**, car configurable en temps réel, à l'exécution du système. Cette phase de transformation dynamique regroupe des transformations structurelles de la trace (suppression d'éléments, tri, et autres opérations paramétrées par l'utilisateur) puis une transcription graphique de la trace obtenue. La vue de cette trace est donc singulière à l'utilisateur tout en étant relative au modèle métier.

4.5. Mise en œuvre

Toutes les traces du système à base de traces, y compris la trace première, sont implémentées conformément au modèle de trace de la figure 8. La sémantique associée à ce modèle est la suivante : une trace est composée de plusieurs actions observables, qui peuvent chacune être locale à l'environnement de l'utilisateur ou partagée par tous les utilisateurs. Chaque action est réalisée par un acteur et impacte (au moins) une entité du système.

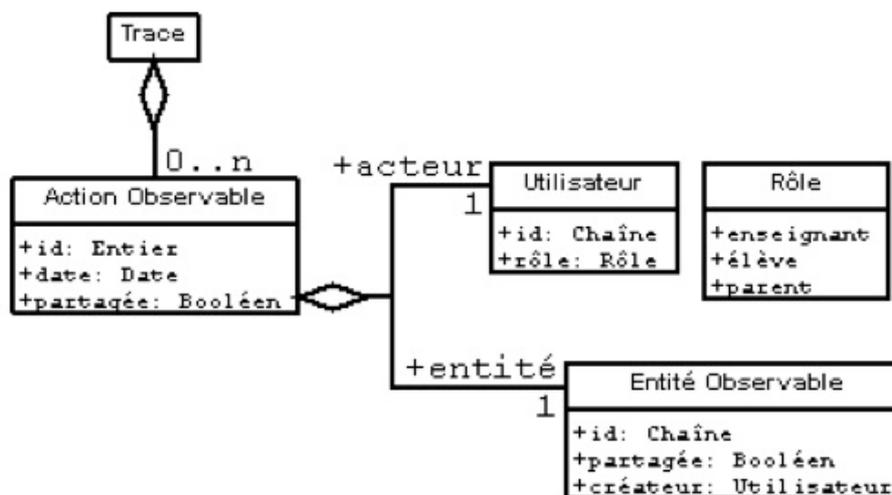


Figure 8. Modèle de trace commun à toutes les traces

Le système de transformation statique comporte des opérations relativement complexes. Pour l'heure, elles sont implémentées en langage Java dans des classes spécifiques. Le système de transformation dynamique qui permet la visualisation graphique interactive est implémenté dans une classe Java générique qui applique les transformations paramétrées par l'objet *Configuration de visualisation*. Les transformations implémentées par ce *Visualiseur* sont : le tri, le filtrage, la suppression, le regroupement ou la fusion d'éléments similaires, le masque visuel. Cette étape de transformation dynamique est générique à toutes les traces ; elle ne souffre pas d'une dépendance à l'activité comme la transformation statique.

5. Exemple et scénario d'application

L'activité pédagogique qui illustre notre système de visualisation interactive de trace est une version très simplifiée d'une activité qui est couramment réalisée en situation réelle avec les élèves d'eLycée, à la différence près que ceux-ci ne disposent pas encore du module de visualisation de traces que nous exposons ici, et que pour maximiser le traçage nous ignorerons la possibilité de converser oralement, la problématique de traçage des communications audio n'ayant pas encore été traitée.

5.1. Description de l'activité de co-traduction d'une planche de bande dessinée

Les élèves sont répartis en binôme. Chaque binôme doit traduire une planche de la bande dessinée "Astérix" de l'anglais vers le français. Ils partagent donc la version originale de cette planche sous forme d'image. Un tableau blanc partagé contient des zones de texte qui représentent les bulles que les élèves remplissent de leurs traductions. Ils peuvent discuter, échanger, et négocier la traduction commune qu'ils affecteront à chaque bulle par l'intermédiaire d'un chat.

5.2. Limites du scénario choisi

Le choix d'une activité collaborative faisant intervenir deux acteurs seulement est assez limitatif en matière de collaboration par rapport à des activités faisant intervenir plusieurs contributeurs. Par exemple, une particularité est que toute action qui n'est pas réalisée par l'un est nécessairement l'œuvre de l'autre. Ainsi, chaque acteur sait qui est l'auteur de chaque action sans avoir besoin de se référer à une assistance comme pourrait l'être la trace.

Nous rappellerons cependant que le but de l'article n'est pas d'analyser les collaborations induites ou assistées par la visualisation réflexive de traces, mais de présenter les moyens mis en œuvre pour permettre cette dernière. Le scénario choisi a uniquement pour but de valider techniquement le système de visualisation de trace et d'illustrer les concepts informatiques présentés dans les sections précédentes.

5.3. Scénario illustré pour deux bulles

5.3.1. Aperçu de l'environnement tracé

Deux acteurs (*Denis* et *Damien*) traduisent deux bulles de cette planche. La [figure 9](#) montre l'état de l'environnement de *Denis* au bout de quelques minutes.

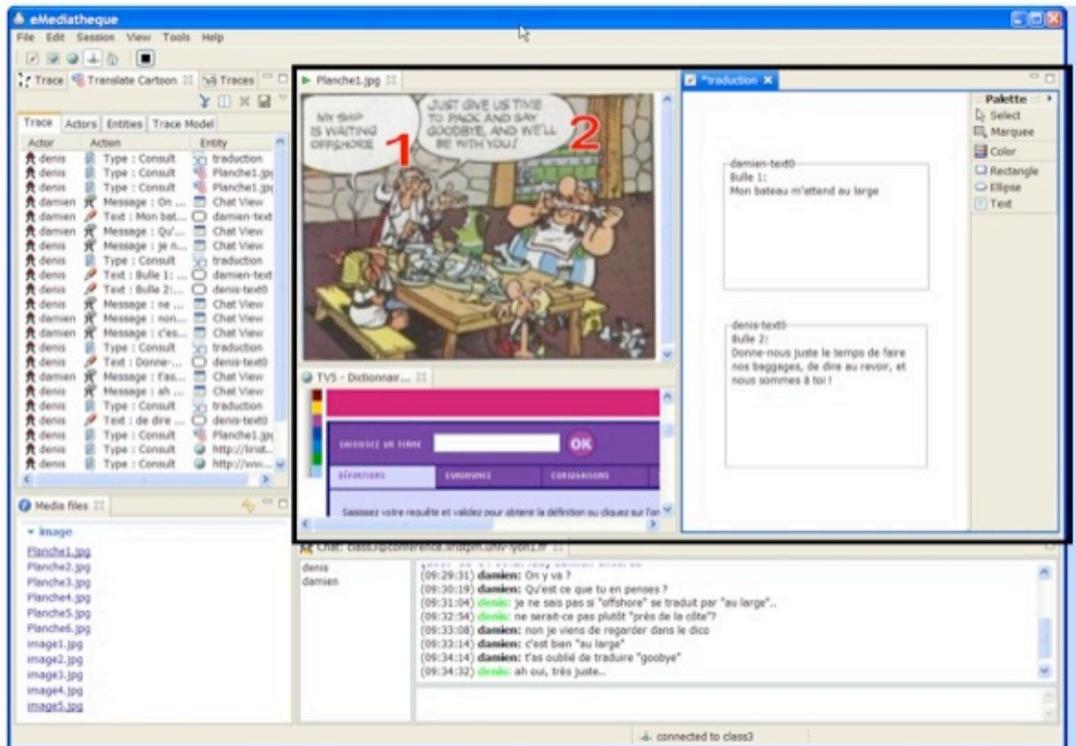


Figure 9. Aperçu de l'environnement au bout de quelques minutes de l'activité

La vue en bas à droite est le *chat*. Le cadre noir contient les éléments partagés par les deux élèves : l'image représentant l'original à traduire, le dictionnaire, et le tableau blanc contenant les zones de texte correspondant aux traductions. En bas à gauche on trouve toutes les ressources média disponibles, et enfin en haut à gauche, la vue sur la trace de l'activité de co-translation.

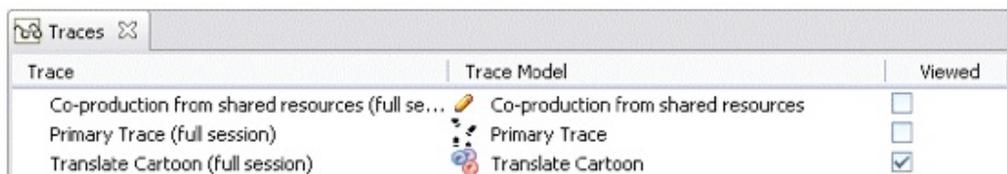


Figure 10. Aperçu de la vue "Traces"

Une autre vue est disponible dans l'environnement mais n'est pas affichée sur cette capture, il s'agit de la vue "Traces" (cf. [figure 10](#)). Cette vue affiche toutes les traces disponibles dans l'environnement, indique le modèle de trace de chacune et précise celles qui sont actuellement vues par l'utilisateur. Sur la [figure 10](#), trois traces sont disponibles : la trace première, la trace d'une activité de co-exploitation de ressources, et la trace d'une activité de "co-translation de bande dessinée", qui est actuellement visualisée par l'utilisateur comme indiqué sur la vue.

5.3.2. Aperçu de la trace

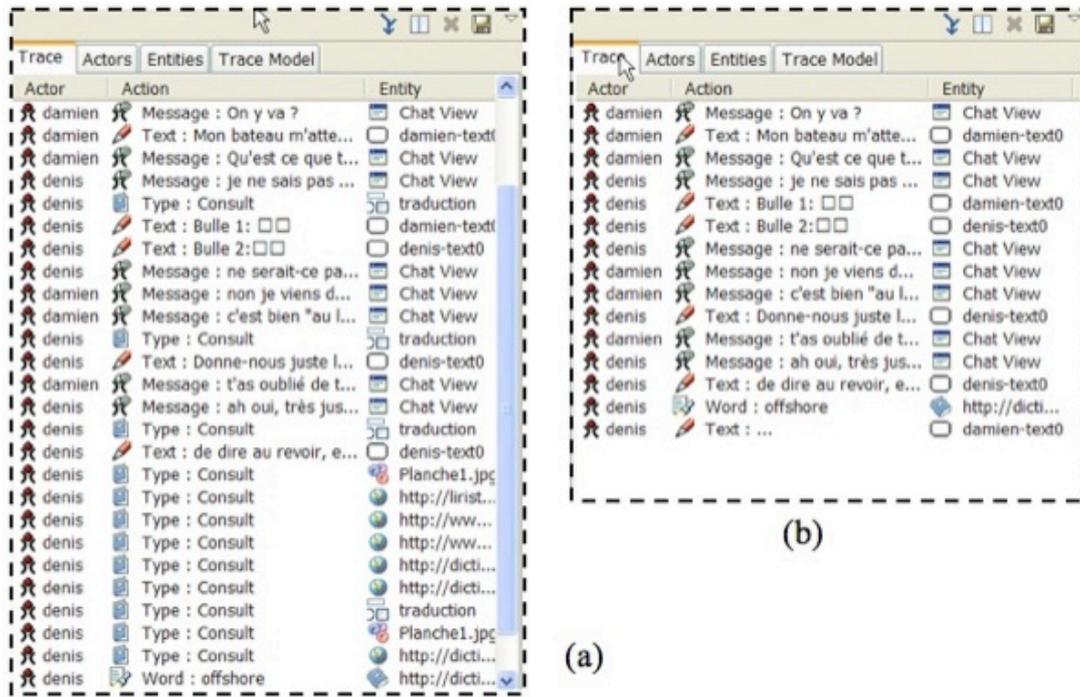


Figure 11. Aperçu de la trace métier de co-traduction de bandes dessinées en version non-filtrée (a) et en version filtrée (b)

Le cadre (a) de la [figure 11](#) montre la trace non filtrée. Chaque ligne représente une action observée de la trace métier. Pour chaque action, nous pouvons voir l'acteur de l'action, le type de l'action réalisée (envoi de message, édition, consultation de ressources ou recherches de mots) et l'entité impactée. Chacun de ces trois éléments peut optionnellement être accompagné d'une icône et d'une couleur. Sur la [Figure 12](#), d'autres onglets sont disponibles. Deux d'entre eux fournissent la liste des acteurs (cadre (a)) qui ont contribué à l'activité et la liste des entités impactées (cadre (b)) avec toutes leurs propriétés. Enfin, un onglet permet de visualiser le modèle de trace de l'activité sous forme d'arbre (cadre (c)). L'apprenant peut utiliser cet onglet pour connaître le descriptif de l'activité (actions possibles, entités possibles et acteurs possibles) et la légende correspondante.

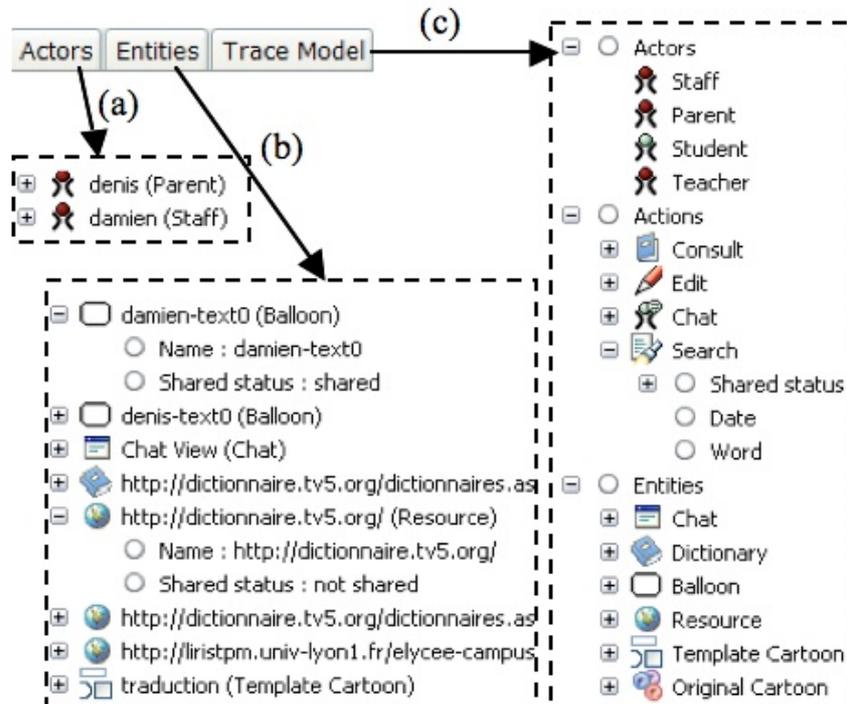


Figure 12 Aperçu des onglets de la vue "trace"

5.3.3. Aperçu des services d'interactions avec la trace

Les menus et la barre d'outils permettent d'effectuer les actions suivantes :

- activation du regroupement de deux actions successives de même type, effectuées par le même acteur et sur la même entité ;
- séparation de la trace en n traces qui représentent chacune la contribution de chaque utilisateur, les fragments de trace générés étant accessibles à la visualisation par la vue "Traces" (cf. [figure 10](#)) ; cette fonctionnalité permet notamment la comparaison graphique des contributions de chacun à l'activité ;
- suppression d'une action observée si l'utilisateur estime qu'elle n'est pas significative ;
- ajout et suppression des règles de filtrage sur les acteurs, actions ou entités ;
- ajout ou suppression de règles de tri des actions ;
- modification du masque visuel des actions, permettant de choisir quels éléments du modèle de trace sont affichés dans l'onglet "Trace" ; l'utilisateur peut par exemple choisir de ne pas afficher les entités s'il juge la vue trop chargée, ou de ne pas afficher d'icône pour représenter les types d'acteur, etc.
- enregistrement de la configuration au format XML.

Dans notre scénario, nous supposons que l'utilisateur estime que la trace est trop polluée par les actions de type *Consult* (consultation d'une ressource), et qu'il ne souhaite plus les voir dans la trace. Il peut installer une règle de filtrage sur les actions de type *Consult* comme le montre la [figure 13](#).

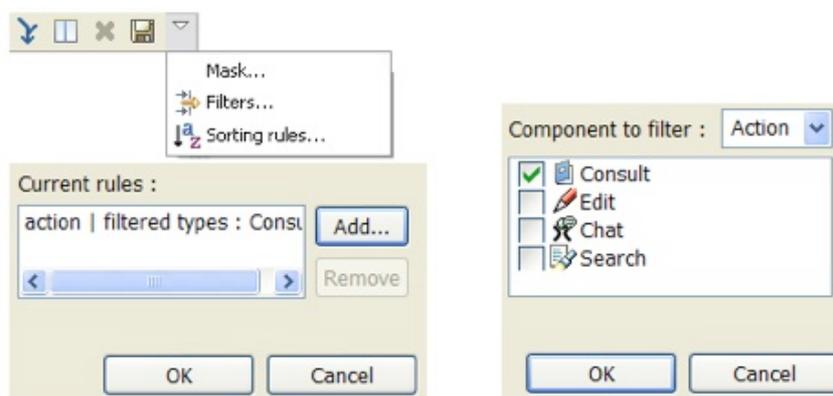


Figure 13. Mise en place d'un filtre sur les actions de type *Consult*

Le cadre (b) de la [figure 11](#) résulte de cette opération de filtrage. Les actions de type *Consult* (consultation des ressources) sont cette fois-ci absentes de la vue. En conséquence la trace est plus légère en actions ; nous pouvons plus facilement y distinguer les éléments clés de l'activité.

6. Extensibilité et généricité de l'approche

6.1. Architecture modulaire en plug-ins

L'architecture logicielle mise en place pour réaliser eMédiathèque s'appuie sur une infrastructure orientée composants permettant de définir l'application comme un assemblage de *plug-ins* qui apportent chacun une contribution au fonctionnement du système. eMédiathèque est ainsi composé : de quatre *plug-ins* (*Cobrowser*, *Media*, *Whiteboard*, *Chat*) implémentant chacun un des outils collaboratifs présentés en section 3.4, d'un *plug-in* central "Client eMédiathèque" qui gère la session de l'élève et l'interopérabilité de tous ces services au sein du même environnement, ainsi que le traçage primitif, et du *plug-in* "SBT" qui écoute la trace première créée par "Client eMédiathèque" et construit les traces métiers liées aux activités en cours et affichées aux utilisateurs.

Dans une application Eclipse RCP, un mécanisme de contribution déclarative des *plug-ins* permet de définir des points d'extension auxquels les *plug-ins* peuvent contribuer. Nous avons ainsi défini un point d'extension "activité / trace métier".

Le premier avantage de cette approche, par rapport à notre problématique de traces visualisées spécifiques aux activités, est que les fonctions de traçage de chaque activité peuvent être fournies par des *plug-ins* séparés (Activité 1, ..., Activité n), indépendants. Chacun de ces *plug-ins* apporte les éléments de modélisation de l'activité et de transformation de la trace première nécessaires à sa visualisation. La section 6.2 détaille la constitution d'un tel *plug-in*. Au final, le *plug-in* "SBT" collecte toutes les activités disponibles et les rend visualisables.

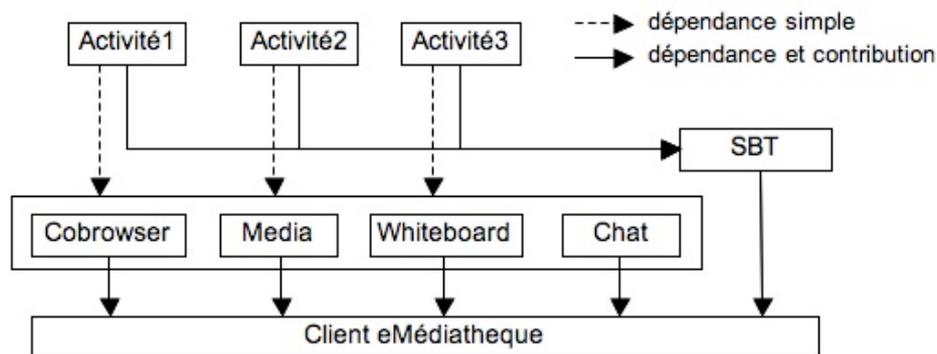


Figure 14. Schéma de contribution et dépendance des *plug-ins* d'eMédiathèque

Le deuxième bénéfice est la possibilité d'ajouter de nouvelles activités, sans nécessité de réinstallation ou de redémarrage d'eMédiathèque : une nouvelle activité rendue disponible sur le serveur sera automatiquement téléchargée et installée ou mise à jour dans eMédiathèque, de façon quasi transparente pour l'utilisateur. Cet effort de genericité et de modularité dynamique dans l'implémentation de la plate-forme permet donc l'introduction et la mise à jour fréquente d'activités tracées, indispensables à toute exploitation et déploiement dans un environnement réel.

6.2. Ajout d'une nouvelle activité tracée

Nous rappelons que la trace première est neutre par rapport à l'activité réalisée et qu'elle doit donc être transformée pour être spécifique à l'activité ajoutée (cf. section 4.2). L'ajout d'une nouvelle activité pédagogique tracée dans eMédiathèque, sous forme de *plug-in*, nécessite trois étapes. Cette procédure ne concerne que la partie *statique* de la transformation de trace, et pas la partie *dynamique*.

6.2.1. Étape 1 : conception du modèle de trace de l'activité

La première étape consiste à réaliser le modèle de trace métier de l'activité à tracer, c'est-à-dire définir les entités et les actions qui font sens dans le cadre de l'activité pédagogique. On pourrait donc également désigner ce *modèle de trace métier* par *modèle d'activité*.

La figure 15 représente le modèle de trace (simplifié) pour l'activité de co-traduction de bandes dessinées et ses relations avec le modèle de trace générique de la figure 8. Trois nouvelles classes d'actions ont été créées par rapport aux classes déjà définies pour le modèle de trace première : la lecture d'une ressource (*Read*), la consultation d'un dictionnaire (*ConsultDict*) et l'édition (*Edit*), par spécialisation des classes d'action de la trace première : *Focus*, *Navigate* et *Modify*. Une nouvelle classe d'entité a été créée : le dictionnaire (*Dictionary*). Il s'agit ici d'une activité qui réutilise et étend beaucoup de classes du modèle de trace première, mais il se pourrait que d'autres activités nécessitent la redéfinition de nouvelles classes d'actions et d'entités, sans rapport avec le panel d'actions de la trace première : un tel modèle de trace spécialiserait donc directement les classe abstraites *Action* et *Entity*.

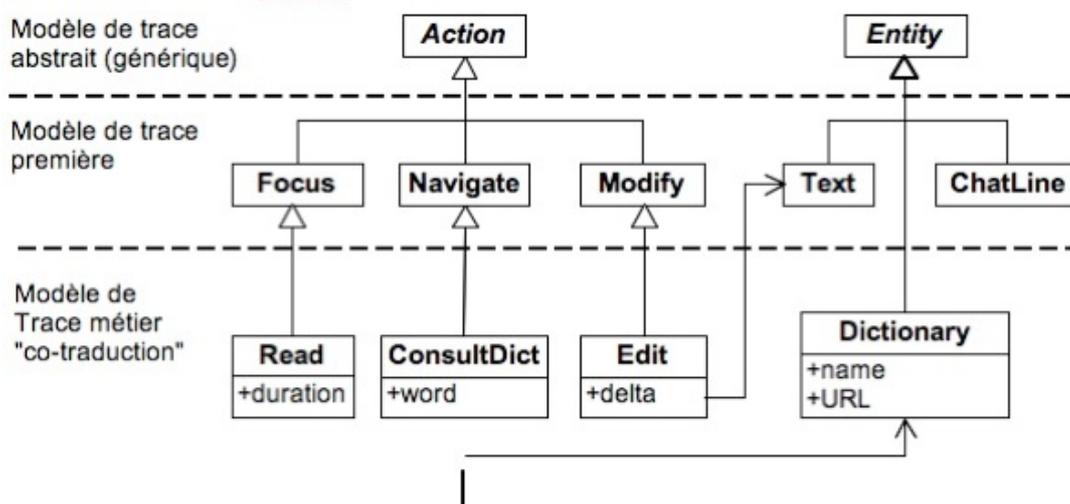


Figure 15. Modèle de trace étendu pour la co-traduction de bandes dessinées (sous forme de diagramme de classes UML)

6.2.2. Étape 2 : réalisation du transformateur

Le transformateur implémente la transformation statique ; son rôle est de construire au fil de l'eau la trace métier à partir de ce que contient la trace première (cf. [figure 7](#)). Il interprète chaque nouvelle action de la trace première et construit éventuellement, en fonction de cette nouvelle action et d'informations de contexte, une action métier significative dans le cadre de l'activité.

Par exemple, lorsque la trace première contient une action de type "modifier" (*Modify*) et que l'entité concernée est une zone de texte (entité de type *Text*), alors le transformateur de l'activité co-traduction crée une action *Edit* dont le paramètre *delta* représentant la chaîne de caractères qui a été éditée est calculé à partir de l'ancienne chaîne de caractère (attribut *oldValue* de l'action de type *Modify*) et la nouvelle (attribut *newValue* de l'action de type *Modify*). Similairement, l'action de lecture (*Read*) correspond à la mise au premier plan d'une page ou d'une ressource pendant une durée supérieure à un certain seuil (action de type *Focus* dans la trace première, augmentée d'un chronométrage).

La charge de définir ces transformateurs revient à une personne ayant la double compétence de pédagogue pour identifier les types d'entités et d'actions significatifs dans l'activité et d'informaticien pour implémenter le modèle de cette activité ainsi que les transformations permettant de transformer la trace première en une trace métier spécifique à cette activité. Nous proposons de le désigner par « concepteur d'activité ». Un éditeur de modèles de trace et de transformations est en cours de développement au LIRIS, mais pour l'heure ces étapes sont réalisées en Java.

6.2.3. Étape 3 : définition des métadonnées de visualisation

Cette étape consiste à affecter à chaque élément du modèle de trace des métadonnées de visualisation qui définissent la correspondance entre les éléments du modèle de trace et des propriétés visuelles (nom, couleur, image). Concrètement, nous associons à chaque élément de la trace (actions, acteurs, entités, et leurs attributs) un descripteur de visualisation. Par exemple, à chaque élément de type *Action* est associé, entre autres, un descripteur (*ComponentTypeInfo*) qui représente le type d'action. Ce descripteur fournit pour ce type d'élément une image, une couleur et un nom. Ces métadonnées fournissent également au mécanisme de visualisation une liste exhaustive de tous les types d'actions, d'entités et d'acteurs possibles dans cette activité, ainsi que leurs attributs. A terme, nous envisageons d'utiliser un mécanisme d'annotation du modèle de trace pour représenter ces métadonnées de visualisation.

7. Conclusion, discussions et perspectives

L'EIAH eMédiathèque est une plate-forme qui a été conçue pour tracer en temps réel et de manière exhaustive l'usage de toutes les fonctionnalités de la plate-forme par l'apprenant, grâce à un mécanisme de réification des actions. Par une *transformation statique* de la trace obtenue configurée au préalable de la session d'apprentissage en ligne, puis chargée dans eMédiathèque, la plate-forme instancie pour chaque type d'activité pédagogique une trace métier dont la sémantique est celle de l'activité qu'elle représente. Tout utilisateur de la plate-forme (apprenant, enseignant, etc.) peut visualiser ses propres traces pendant le cours de l'activité et effectuer des opérations basiques sur ces traces comme le tri des éléments qu'elle contient, le regroupement, le filtrage, et le mode d'affichage des propriétés ou entités (les masques visuels) ; nous appelons ces opérations les *transformations dynamiques*.

La mise en œuvre du traçage d'une nouvelle activité se matérialise par l'ajout d'un *plug-in* qui se greffe au système de visualisation en apportant trois éléments : le modèle de trace métier de l'activité, les règles de transformation à appliquer à la trace première pour obtenir une trace de ce modèle, et les métadonnées qui permettent d'afficher graphiquement les propriétés de la trace métier (noms, images et couleurs).

Il convient de préciser que les mécanismes de traçage et de visualisation interactive de la trace pour la réflexivité peuvent s'appliquer à d'autres types d'activités, qui ne soient ni forcément dédiées à l'apprentissage, ni forcément collaboratives. Les concepts abordés dans cet article ont majoritairement été traités du point de vue de leur faisabilité et de leur réalisation informatique, toutefois les perspectives ouvertes par un tel système sont nombreuses.

7.1. Vers une assistance réflexive à base de traces

Nous avons évoqué la réflexivité dès l'introduction car c'était l'objectif initial du projet et notre système a été bâti selon l'hypothèse qu'une visualisation par l'apprenant de ses traces d'activité personnelles favoriserait son activité pédagogique. Le besoin "d'assistance métacognitive" à l'utilisateur d'un système est de plus en plus communément reconnu, particulièrement pour les applications collaboratives ([Dimitracopoulou et Bruillard, 2006](#)). Dans l'approche qu'un utilisateur peut se faire d'un dispositif, ([Richard, 1983](#)) distingue deux logiques qu'il peut mettre en œuvre. La *logique de fonctionnement* s'applique lorsque l'utilisateur connaît le résultat de chaque commande du dispositif et agit en fonction de cette connaissance du fonctionnement du dispositif. La *logique d'utilisation* régit ce que l'utilisateur sait effectuer à l'interface du dispositif comme suite d'actions pour obtenir le comportement souhaité, sans forcément en comprendre le fonctionnement interne. Dans la même lignée, ([Mille et Prié, 2006](#)) parlent de *logique de conception* pour désigner l'usage du système telle que prévu par le concepteur et de *logique*

d'utilisation pour désigner l'usage réel ou désiré par l'utilisateur. Les auteurs expliquent qu'il existe dans tout système un hiatus entre la logique de conception et la logique d'utilisation et défendent que présenter systématiquement un retour à l'utilisateur de ce qu'il fait et de l'impact qu'il a sur son environnement permettra de réduire ce hiatus.

La réflexivité dans eMédiathèque peut s'effectuer à trois niveaux : la réflexivité de l'apprenant en temps réel par visualisation de ses traces pendant son activité ; la réflexivité de l'apprenant *a posteriori* par visualisation des traces d'une activité passée ; et ce que nous désignons ici par "réflexivité de groupe" concernant la possibilité pour un participant à l'activité de se positionner par rapport à celle du groupe auquel il appartient par consultation des traces. A l'heure actuelle, nous n'avons pas pu confirmer ou infirmer ces hypothèses de manière scientifiquement fondée. Cependant nous disposons de deux retours d'expérience sur notre système.

Le premier provient d'une expérimentation réalisée sur eMédiathèque par des membres de notre équipe de recherche. Chaque participant s'est attelé à l'activité décrite en section 5.1 en binôme. Le but de cette expérimentation était de valider techniquement le prototype développé, de susciter des remarques spontanées de la part des participants sur les fonctionnalités du système, l'ergonomie, et de préparer le terrain pour des expérimentations futures "grandeur nature", réalisées avec des étudiants en réelle situation d'apprentissage, en vue d'anticiper les difficultés éventuelles de leurs mises en place et de savoir orienter les activités vers des situations permettant d'observer des usages réflexifs de la trace.

Le deuxième retour d'expérience dont nous disposons provient cette fois des étudiants participant aux sessions en classes virtuelles de la société eLycée. Dans la version d'eMédiathèque qu'ils ont à leur disposition, les étudiants ne peuvent pour l'instant visualiser que leurs traces premières (cf. [figure 4](#)), censées être non facilement réutilisables dans le contexte d'une activité pédagogique. Nous avons pu constater des situations de réutilisation de la trace élémentaire, comme le rechargement d'entités précédemment fermées dans l'espace ou la remise en contexte temporelle de certaines actions réalisées, grâce à l'information temporelle des actions.

Les prochains travaux sur la réflexivité du projet relèveront des sciences cognitives et traiteront les problématiques suivantes : quelles sont les situations favorables à la réflexivité lors d'une activité pédagogique collaborative ? Quels sont les apports de la réflexivité dans les processus d'apprentissage ? Comment mesurer le "taux de réflexivité" d'une activité ?

7.2. Vers un partage d'expérience à base de traces documentarisées

Nous entendons par *documentarisation* d'épisodes de la trace le fait de donner à un épisode de trace (généralement métier), défini et visualisé par un utilisateur, le statut de document à part entière, au même titre qu'une autre ressource mise à disposition de l'utilisateur par la plate-forme (tableau blanc, vidéo, etc.). Un tel épisode de trace peut alors être visualisé, partagé, annoté, et faire l'objet de requêtes de recherche dans une base d'épisodes de traces.

eMédiathèque permet d'ores et déjà de sauvegarder un épisode de trace. En effet, la *trace vue* par l'utilisateur est le résultat d'un couplage par le *Visualiseur* de la *trace métier* et d'une certaine *Configuration de visualisation*, la trace métier étant elle-même élaborée à partir de la *trace première* et d'un *Transformateur* dédié à l'activité (cf. [figure 7](#)). Ainsi la donnée de la *trace première* et de la *Configuration de visualisation* permet de restituer la *trace vue*. eMédiathèque persiste donc un épisode de *trace vue* en sauvegardant l'épisode de *trace première* correspondant et la *Configuration de visualisation*.

Cette séparation de la présentation (objet *Configuration de visualisation*) et du contenu (*trace première*) permet à différents utilisateurs de la plate-forme de visualiser différents épisodes sous le même point de vue, mais également de visualiser le même épisode sous différents points de vue. En pratique, on peut imaginer qu'un enseignant visualise un épisode de trace d'un étudiant préalablement mis en partage pour mieux situer ses difficultés et lui donner une meilleure assistance. Un système de partage et d'échange d'épisode est à implémenter dans eMédiathèque, mais il nous faut d'abord ajouter des possibilités d'annotation d'épisodes et de recherche sur des bases de traces, qui feront eux aussi l'objet de recherches futures.

7.3. Vers un "awareness collaboratif" à base de traces

La portée "apprentissage en situation collaborative" découle naturellement du fait que la plate-forme applicative abordée dans cet article est une plate-forme collaborative synchrone pour l'apprentissage. La première conséquence est que les traces sous-jacentes à la visualisation de traces comportent une dimension collaborative. En effet, la trace présente dans l'environnement eMédiathèque de chaque utilisateur comporte deux types d'action. Les *actions locales* sont celles qu'il a réalisées dans son environnement sans impacter les ressources partagées. Par exemple, naviguer sur Internet peut se faire en session locale. Les *actions partagées* sont celles qui s'effectuent sur des ressources partagées, comme les tableaux blancs, les sessions Web partagées, etc. Dans ce cas, les actions réalisées apparaissent dans les traces de chacun.

De cette manière, eMédiathèque contribue à la problématique de l'*awareness* dans les environnements collaboratifs distribués en temps réel. L'*awareness* collaboratif peut se définir comme la compréhension en temps réel du travail effectué par les participants sur l'espace partagé : d'où ils travaillent, de ce qu'ils font, de leurs futures intentions ([Gutwin et. al. 1996b](#)). Par exemple, dans leur système en vues "Radar", ([Gutwin et. al. 1996a](#)) portent le focus de l'espace de travail partagé sur certaines zones précises par des rectangles spécifiant quel participant travaille sur la zone focalisée, ce qu'il est en train de voir de l'espace partagé par des vues résumées et des données personnelles le

concernant (nom, photo...). La trace est certainement une source riche en information pour aider à bâtir des systèmes d'*awareness* collaboratif mais nous pensons que la trace elle-même, visualisée par les utilisateurs à travers le système que nous avons présenté ici, peut contribuer à cet *awareness*.

Prenons l'exemple d'une activité d'un élève de la classe en train de regarder une vidéo ou d'effectuer une autre action pendant qu'un deuxième élève écrit sur le tableau blanc partagé. Dans ce cas l'apparition d'une action de type "Ecrire" dans la trace du premier élève, dont l'auteur est le deuxième élève, lui permet d'être informé que la production commune aux élèves de la classe est en train d'évoluer sans lui. Les expérimentations prévues sur cette plate-forme devront également permettre d'établir la contribution de notre système de visualisation interactive de traces à l'*awareness* collaboratif et d'observer dans quelles mesures l'élève est conscient des contributions effectuées par chacun des autres participants de la classe par la trace.

BIBLIOGRAPHIE

AVOURIS N., KOMIS V., FIOTAKIS G., MARGARITIS M., VOYIATZAKI E. (2005). Logging of fingertip actions is not enough for analysis of learning activities. *Usage analysis in learning system AIED'05 Workshop*.

CHAMPIN P.-A. (2003). ARDECO: an assistant for experience reuse in Computer Aided Design. Proceedings of WS 5 of ICCBR'03: From structured cases to unstructured problem solving episodes. Trondheim (NO), NTNU, Trondheim, p. 287-294.

CHERNOFF H. (1973). The Use of Faces to Represent Points in k-Dimensional Space Graphically. *Journal Amer. Statistical Association*, Vol. 68, p. 361-368.

DIMITRACOPOULOU A., BRUILLARD E. (2006). Enrichir les interfaces de forums par la visualisation d'analyses automatiques des interactions. revue STICEF : *revue francophone de Sciences et Technologies de l'Information et de la Communication pour l'Éducation et la Formation*, Vol. 13, p. 345-398.

DOURISH P., BELLOTTI V. (1992). Awareness and Coordination in Shared Workspaces. In *proceedings of the 1992 ACM conference on Computer-Supported Cooperative Work (CSCW '92)*. Toronto, Ontario, Canada. p. 107-114.

eLycée Teaching Method.

http://www.elycee.com/elycee-campus/v2/teaching_method.php (consulté le 30 avril 2007).

FRANCE L., HERAUD J.-M., MARTY J.-C., CARRON T., HEILI J. (2007). Monitoring Virtual Classroom: Visualization Techniques to Observe Student Activities in an e-Learning System. *Proceeding of the 6th IEEE International Conference on Advanced Learning Technologies*. Kerkrade, The Netherlands, p. 716-720.

GEORGEON O., MILLE A., BELLET T. (2006). Abstract: un outil et une méthodologie pour analyser une activité humaine médiée par un artefact technique complexe. *Actes de Ingénierie des Connaissances IC 2006*, Nantes.

GUTWIN C., GREENBERG S., ROSEMAN M. (1996). Workspace Awareness Support with Radar Views. In M. Tauber (ed.). *Companion Proceedings of CHI'96: Human Factors in Computing Systems (CHI'96 Conference Companion)*, ACM Press, p.210-211.

GUTWIN C., GREENBERG S., ROSEMAN M. (1996). Workspace Awareness in Real-Time Distributed Groupware: Framework, Widgets, and Evaluation. *Proceedings of HCI on People and Computers XI*, Springer-Verlag, London, UK, p. 281-298.

HERAUD J.-M., FRANCE L., MILLE A. (2004). Pixed: An ITS that guides students with the help of learners' interaction logs. *7th International Conference on Intelligent Tutoring Systems (Workshop Analyzing Student-Tutor Interaction Logs to Improve Educational Outcomes)*. Maceio, Brésil. p. 57-64.

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

HILBERT D.-M., REDMILES D.-F. (2000). Extracting Usability from User Interface Events. *ACM Computing Surveys*. Vol. 32, No 4, p. 384-421.

IKSAL S., CHOQUET C. (2005). Usage Analysis Driven by Models in a Pedagogical Context. *Workshop Usage analysis in learning systems at AIED2005: 12th International Conference on Artificial Intelligence in Education*. Amsterdam, Pays-Bas, p. 49-56.

LOHSE G.-L., BIOLSI K., WALKERS N., RUETER H. (1994). *A classification of visual representations*. Communication of the ACM, Vol. 37, No 12, p 36-49.

MARTY J., HERAUD J., CARRON T. & FRANCE L. (2007). Matching the Performed Activity on an Educational Platform with a Recommended Pedagogical Scenario: A Multi-Source Approach. *Journal of Interactive Learning Research*. 18 (2), Chesapeake, Etats-Unis, p. 267-283.

MASCIOTRA D. (2005). Réflexivité, métacognition et compétence. *Vie pédagogique* 134. p. 29-31.

MAZZA R., DIMITROVA V. (2003). CourseVis: Externalising Student Information to Facilitate Instructors in Distance Learning. *Proceedings of the International conference in Artificial Intelligence in Education*. U.Hoppe, F. Verdejo, J. Kay (eds.), Sydney, Australia, p. 279-286.

MILLE A., PRIÉ Y. (2006). Une théorie de la trace informatique pour faciliter l'adaptation dans la confrontation logique d'utilisation/logique de conception. *13eme Journées de Rochebrune - Traces, Enigmes, Problèmes : Emergence et*

construction du sens - Rencontres interdisciplinaires sur les systèmes complexes naturels et artificiels. Rochebrune, France, p. 12.

MOLLO V. (2002). La construction des procédures par la pratique : le rôle des outils ergonomiques. *Actes du XXXVIIIème Congrès de la SELF, "Les évolutions de la prescription"*. Aix-en-Provence, France, p. 201-208.

NOVICK L.-R., HURLEY S.-M. (2001). To matrix, Network, or Hierarchy : That is the Question. *Cognitive Psychology*. Vol. 42, p. 158-216.

OLLAGNIER-BELDAME M. (2006). *Traces d'interactions et processus cognitifs en activité conjointe : Le cas d'une co-rédaction médiée par un artefact numérique*. Rapport de thèse en sciences-cognitives de l'Université Louis Lumière Lyon 2.

PAPERT S. (1993). *The Children's Machine : Rethinking School in the Age of the Computer*. Basic Books, New York.

PLAISANT C., MILASH B., ROSE A., WIDO S., SHNEIDERMAN B. (1996). 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.

PLAISANT C., ROSE A., RUBLOFF G., SALTER R., SHNEIDERMAN B. (1999). 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.

RICHARD J.F. (1983). *Logique de fonctionnement et logique d'utilisation*. Rapport de recherche INRIA, n° 202, 51 p.

ROSSI F., LECHEVALLIER, EL GOLLI A. (2005). *Visualisation de la perception d'un site Web par ses utilisateurs*. Suzanne Pinson and Nicole Vincent editors, Actes EGC 2005, RNTI-E-3, Vol. 2, p 563-574, Cépaduès-Éditions, Paris, 2005.

ROUSSEL N., TABARD A., LETONDAL C. (2006). All you need is log. *WWW 2006 Workshop on Logging Traces of Web Activity: The Mechanics of Data Collection*, 4 pages.

SANDERSON P.-M., FISHER C.-A. (1994). *Exploratory sequential data analysis: foundations*. *Human-Computer Interaction*. Vol. 9(3), p 251-317.

SÉJOURNÉ A., BAKER M., LUND K., MOLINARI G. (2004). 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.

SETTOUTI L.-S., PRIE Y., MILLE A., MARTY J.-C. (2007). Vers des Systèmes à Base de Traces modélisées pour les EIAH. Soumis au numéro spécial "Analyses des traces d'utilisation dans les EIAH" de la revue *STICEF, revue francophone de Sciences et Technologies de l'Information et de la Communication pour l'Éducation et la Formation*.

WEXELBLAT A., MAES P. (1999). Footprints: History-rich tools for information foraging. *Proceedings of ACM CHI 99 Conference on Human Factors in Computing Systems*, p. 270-277.

¹ <http://www.elycee.com/>

² <http://www.marratech.fr/>

³ <http://www.illuminate.com>

⁴ <http://www.wimba.com>

⁵ <http://www.adobe.com/products/breeze/>

⁶ <http://www.marratech.com>

⁷ <http://www.ilinc.com>, en particulier le produit LearnLinc

⁸ <http://blackboard.com/>

⁹ <http://moodle.org/>

¹⁰ <http://cluster-isle-eiah.liris.cnrs.fr/>

¹¹ Une commande étant une action ré exécutable, et dans certains cas annulable ; dans le cadre de cet article, la différence entre action et commande n'est toutefois pas significative.

¹² voir la page d'accueil du projet EMF pour plus de détails sur ces différentes technologies : <http://www.eclipse.org/modeling/emf/>

■ A propos des auteurs

Damien CRAM est doctorant au laboratoire LIRIS depuis Avril 2007 au sein de l'équipe SILEX (Supporting Interactions by Learning from Experience), animée par Alain MILLE. Il a récemment été diplômé de l'Ecole Centrale de Lyon et d'un Master Recherche en Informatique en Intelligence Artificielle et a réalisé son stage de fin d'études sous l'encadrement de

Denis JOUVIN et d'Alain MILLE, dans le cadre du projet eLycée. Son projet de recherche consistait à étudier la visualisation des traces d'interactions en temps réel pour la réflexivité et à en développer un prototype sur l'EIAH eMediatheque.

Adresse : LIRIS UMR 5205 , Université de Lyon, Université Claude Bernard, Bâtiment Nautibus (710), 43, Boulevard du 11 Novembre 1918, 69622 VILLEURBANNE CEDEX France

Courriel : damien.cram@liris.cnrs.fr

Toile : <http://liris.cnrs.fr/damien.cram>

Denis JOUVIN est depuis deux ans directeur technique et ingénieur R&D de eLycée S.A.S., une entreprise d'enseignement à distance du français, et parallèlement post-doctorant au laboratoire LIRIS, dans l'équipe SILEX (Supporting Interactions by Learning from Experience), où il travaille notamment sur l'usage des traces dans les plateformes collaboratives synchrones orientées e-learning. Après avoir obtenu le diplôme d'ingénieur INSA en informatique, il a effectué son doctorat en informatique sur une thématique Systèmes Complexes et Systèmes Multi-Agents, à l'université Claude Bernard Lyon 1, et en collaboration avec le laboratoire d'intelligence artificielle (AI Lab) du KAIST, en Corée du Sud. Il a par ailleurs travaillé plusieurs années à l'étranger (Corée du Sud, Laos, Vietnam) comme ingénieur informatique.

Adresse : LIRIS UMR 5205 , Université Claude Bernard Lyon 1, Bâtiment, Nautibus, 43 Boulevard du 11 Novembre 1918, 69622 Villeurbanne Cedex, France

Courriel : denis.jouvin@liris.cnrs.fr

Toile : <http://liris.cnrs.fr/denis.jouvin>

Alain MILLE est professeur à l'université Lyon1, chercheur au LIRIS UMR CNRS 5205. Informaticien, il s'intéresse à l'ingénierie de l'expérience tracée et en particulier à l'exploitation de cette expérience tracée pour faciliter les usages TIC. Animateur de l'équipe SILEX (Supporting Interactions by Learning from Experience) de l'axe thématique COSYCO (Connaissance et Systèmes Complexes) du laboratoire, il est responsable également d'une structure de recherche pluridisciplinaire (PPF Apprentice) spécialisée sur la question de l'apprentissage avec les TICE. Dans le domaine des EIAH, il s'intéresse spécifiquement aux usages réflexifs de l'expérience tracée facilitant l'appropriation des différentes facettes EIAH et le partage d'expérience dans le cadre de l'apprentissage collaboratif.

Adresse : LIRIS UMR 5205 , Université de Lyon, Université Claude Bernard, Bâtiment Nautibus (710), 43, Boulevard du 11 Novembre 1918, 69622 VILLEURBANNE CEDEX France

Courriel : alain.mille@liris.cnrs.fr

Toile : <http://liris.cnrs.fr/alain.mille>

Référence de l'article :

Damien Cram, Denis Jouvin, Alain Mille, Visualisation interactive de traces et réflexivité : application à l'EIAH collaboratif synchrone eMédiathèque, *Revue STICEF*, Volume 14, 2007, ISSN : 1764-7223, mis en ligne le 03/03/2008, <http://sticef.org>

© Revue Sciences et Technologies de l'Information et de la Communication pour l'Éducation et la Formation, 2007