



République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études

pour l'obtention du diplôme de Master en Informatique

Option: Système d'Information et de Connaissances (S.I.C)

Thème

Génération du profil d'apprenant à partir de traces modélisées de Moodle

Réalisé par :

- HAFFAF Soufyane
- ZENATI MOHAMMED El Amine

Présenté le 29 Juin 2013 devant le jury composé de MM.

- BENAMAR Abdelkrim (Président)
- HALFAOUI Amel (Encadrant)
- BELABED Amine (Examineur)
- BOUDEFLA AMINE (Examineur)
- BENDIABDELLAH M^{ed} Hakim (Co-Encadrant)

Année universitaire : 2012-2013

Remerciements

Avant toute chose, on remercie Dieu le tout puissant de nous avoir donné la force, la patience et le courage d'entreprendre la réalisation de ce mémoire.

Et c'est avec une immense reconnaissance; que nous adressons nos remerciements les plus sincères à notre encadreur «Mme HALFAOUI Amel», qui nous a inculqué des bases solides en la matière et nous a entre autre donné le courage nécessaire afin d'arriver à accomplir cette tâche.

On lui présente donc tous nos sentiments de gratitude.

Nous tenons à témoigner de notre enthousiasme et remercions ceux et celles qui nous ont prêtés attention, aide et assistance à savoir «Mr BELABED Amine» de qui on a sollicité une aide, « Mr Bendiabdellah M^{ed} Hakim » pour des finitions et notre ami « Sami » qui nous a aidé lors de la rédaction.

Nous témoignons entre autre notre reconnaissance à tous nos professeurs du département d'informatique, actuels et antérieurs ; qui ont grandement contribués à notre progression, ainsi qu'à tous les étudiants de notre département auxquels nous adressons nos salutations les plus sincères.

Sofyane & Amine

Dédicaces

A mon père, el-hadj

En toi mon père ; je vois un père dévoué à sa famille. Ta présence en toute circonstance m'a maintes fois rappelé les sens de la responsabilité.

A ma mère, Zahia

En toi ; je vois la maman parfaite ; toujours prête à se sacrifier pour le bonheur de ses enfants.

Merci pour tout.

A mes oncles et mes tantes, à mon frère et ma sœur, je vous dédie ce travail pour qui ma réussite est très importante. Que dieu vous bénisse pour tout ce dont vous avez fait pour moi.

A mes chers amis, je commence par mon binôme Sofyane et sa famille, A Sami, Younes, Amine, Nadir, Imane, Meriem, Sarah et pokémon Ismahan

Amine

Dédicaces

Je commence par rendre grâce à dieu et sa bonté, pour la patience, la compétence et le courage qu'il m'a donné pour arriver à ce stade.

Avec mon énorme plaisir, un cœur ouvert et une joie immense, que je dédie ce modeste travail tout d'abord ; à mes parents pour leurs amour leurs sacrifices, et leurs encouragements qui ont fait de moi ce que je suis aujourd'hui.

A ma sœur Radia qui n'a jamais cessé de m'encourager pour qui je souhaite tout le bonheur pour l'heureux évènement qui l'attend

A une personne très chère à mon cœur qui m'a toujours soutenu et m'encourager.

A mon binôme Amine avec qui j'ai partagé les joies et les difficultés durant ce projet.

A mon cher oncle Sidahmed mes tantes ainsi qu'à mon cousin

Mohammed et ma cousine Asma.

A mes amis et frères Mahmoud, Anes Sofiane Ali et Omar ainsi qu'à Oussama, Yasser, Adim, Hakim et Mustapha, sans oublier Imane Meryem et Ismahan et tous mes amis sans exception

Sofyane

Résumé

Ce mémoire s'inscrit dans le domaine de l'exploitation des traces dans les principaux systèmes d'apprentissages, il s'intéresse aux plateformes d'e-learning en particulier « Moodle ». Le système proposé du nom de « GenProApp » aura pour but la génération de profil des apprenants en sessions d'apprentissage sur la plate-forme e-learning. Pour y parvenir, nous avons proposé un modèle de « traces premières » sous forme d'une ontologie OWL, ce travail sollicitera un processus qui prend en charge les collectes de données, les modélisations des traces brutes selon un modèle ainsi qu'une génération de profil qui obéit à une norme d'évaluation des apprenants.

Mots clés : Traces première, Trace modélisée, Calcul des indicateurs, Génération du profil.

Abstract

This research work is in the field of exploitation of traces in fundamental learning systems; this work focuses on an e-learning platform which is "Moodle". The software "GenProApp" will aim to generate profiles of learners on learning platform sessions. To achieve this, we proposed a model of "raw traces" in the shape of an OWL ontology, this work will solicit a process that supports data collection, modeling raw traces following a model and a generation of profiles that follows a standard evaluation of learners.

Keywords: Raw traces, Traces modeled, Calculating indicators, Profile generation.

ملخص

هذه المذكرة تدرج في مجال استغلال الآثار في أنظمة التعلم الرئيسية، و تهتم بمنصات التعلم الإلكتروني على وجه الخصوص « موودل ». النظام المقترح في هذه المذكرة « GenProApp » يهدف لإنشاء ملفات تعريف للمتعلمين أثناء دورات على منصة التعلم الإلكتروني. لتحقيق هذا، اقترحنا نموذجاً للآثار الأولية على شكل أنطولوجيا OWL، هذا العمل يتبع العمليات التي تدعم جمع البيانات، نمذجة الآثار الأولية باستخدام نموذج وإنشاء ملفات تعريف للمتعلمين الذي يتبع ل التعريف الذي يتبع معيار تقييم للمتعلمين.

كلمات البحث: آثار أولية، آثار نمذجة، حساب المؤشرات، استخراج الملف الشخصي.

Table des matières

Introduction générale	3
Contexte du travail	3
Plan de mémoire.....	4
Chapitre I Etat de l’art	5
I. Introduction.....	6
II. Définitions	6
II.1.E-learning	6
II.2.Plate-forme de E-learning.....	6
II.3.Trace	7
II.4.Modèle de traces	7
II.5.Trace modélisée.....	8
II.6.Indicateur	8
III. Quelques travaux sur les E.I.A.H. et les traces.....	8
III.1.Visualisations des traces	8
III.1.1. E-Médiathèque	8
III.1.2. CourseVis	9
III.1.3. ClassroomVis	10
III.2.E.I.A.H. et expérimentations pour recueillir les traces	11
III.3.Synthèse	13
IV. Reformulation de la problématique	15
Chapitre II Modélisation des traces brutes de Moodle	16
I. Introduction.....	17
II. Analyse des données de Moodle (traces).....	17
III. Modélisation des traces.....	21
III.1.Modèle de trace première.....	21
III.2.L’ontologie TrMood.....	23
IV. Formalisation des règles de calculs des indicateurs	27
IV.1.Calcul d’indicateurs	27
IV.1.1.Mode « par défaut »	28
IV.1.2.Mode « avancé »	28
IV.2.Echelle de notation.....	30

V. Conclusion	31
Chapitre III Génération du profil d'apprenant.....	32
I. Introduction.....	33
II. Architecture conceptuelle	33
III. Modélisation du système	34
III.1.Diagramme de cas d'utilisation.....	34
III.2.Diagramme de séquence	34
III.3.Diagramme de classe.....	35
IV. Processus de génération à partir des traces modélisées	36
V. Cas d'application	38
VI. Conclusion	44
Conclusion générale.....	45
Références Bibliographiques	48

Introduction générale

Contexte du travail

L'utilisation de l'informatique pour l'apprentissage et l'enseignement se développe et évolue sous le coup des différents facteurs inter-reliés ; à titre d'exemple on cite une poussée technologique ; à savoir un faible coût des technologies ainsi qu'une facilité lors d'un ou plusieurs usages, l'évolution des connaissances scientifiques, la demande sociale ou encore l'évolution des pratiques des enseignants et des apprenants.

Cette évolution a notamment favorisé l'apparition du concept d'E.I.A.H. (Environnements Informatiques d'Apprentissage Humains) qui est en soi un environnement informatique conçu dans le but de favoriser l'apprentissage humain, c'est-à-dire la construction de connaissances chez un apprenant [1].

Un des aspects de la recherche en E.I.A.H. concerne la modélisation des connaissances de l'apprenant, dans le but d'avoir une vision la plus exacte possible de ce qui a été assimilé. L'implémentation de tels modèles s'effectue à partir de traces d'interactions entre l'élève et son milieu ; que cette interaction ait lieu avec le professeur ou avec un logiciel dédié à l'enseignement d'une matière spécifique.

Il est en effet théoriquement facile de « garder trace » de l'activité pour l'analyser afin d'élaborer des « indicateurs d'activité » permettant une adaptation de l'E.I.A.H. par l'enseignant et/ou par l'E.I.A.H. (lui-même) selon des règles prédéfinies lors d'une personnalisation.

Le travail présenté dans ce mémoire s'intéresse à l'exploitation des traces de la plateforme Moodle pour évaluer et générer le profil de l'apprenant, il s'articule autour de trois principales étapes :

La première concerne « la conception d'une ontologie OWL :TrMood » servant à modéliser les traces laissées par Moodle.

La deuxième se rapporte à « récolter les traces brutes issues d'une session d'apprentissage » pour construire une trace première modélisée basé sur l'ontologie « TrMood » puis calculer des indicateurs servant à générer le profil de l'apprenant.

La troisième et dernière étape « présente le système GenProApp » la génération du profil d'apprenant qui est une application java qui exploite l'ontologie TrMood et utilise des règles SPARQL pour le calcul des indicateurs.

Plan de mémoire

Ce mémoire quant à lui se structure comme suit :

Un chapitre traitant « l'état de l'art et les travaux relatifs à notre contexte de travail ».

Un second chapitre concerne « la collecte » ainsi que « la modélisation des traces et le calcul d'indicateurs » nous présentons aussi l'ontologie que nous avons élaboré « TrMood » qui est un modèle de traces de Moodle.

Enfin, le dernier chapitre quant à lui se consacre à définir notre système « GenProApp » qui consiste en une application qui a partir des trace de Moodle génère le profil d'apprenant.

Chapitre I

Etat de l'art

I. Introduction

Comme il a été cité dans le plan du mémoire, nous entamons ce chapitre par l'état de l'art sur les différents travaux en relation avec la modélisation de traces et le calcul des indicateurs d'interactions. Au cours de cet état de l'art, nous reprenons les définitions des plateformes de E-Learning, l'E.I.A.H., traces, indicateurs, nous étudions les travaux en relations, nous ferons une synthèse de ces travaux et nous termineront par poser notre problématique.

II. Définitions

II.1. E-learning

Signifie "enseignement électronique"; c'est une discipline où théorie d'enseignement et technologies des réseaux informatiques sont combinées pour permettre à des apprenants de réaliser leurs cursus par l'intermédiaire d'un réseau informatique (Internet ou Intranet) [2].

II.2. Plate-forme de E-learning

Une plate-forme de E-learning est un LCMS est un système pour créer et gérer de la matière pédagogique pour la formation mixte (à distance ou non). C'est tout d'abord un espace qui centralise le dépôt de contenus d'apprentissage, afin qu'ils puissent facilement être recherchés, identifiés et réutilisés à tout moment, pour n'importe quel besoin de formation.

Le LCMS intègre ensuite des outils de création de contenus de formations, manipulés par les équipes de conception pédagogique et non par les apprenants [3]. Ces systèmes ont pour ambition de mettre en ligne des enseignements complets où l'étudiant peut préparer son ingénieur, sa licence, son bac, ... via ces plates-formes.

Les principales fonctionnalités d'une plate-forme sont :

- Création de cours et de tests.
- Gestion de documents pédagogiques (indexation, classification, mises à jour, ...).
- Gestion d'un espace de travail coopératif entre apprenants et/ou entre enseignants.
- Suivi de l'apprentissage de l'apprenant et son évaluation.
- Gestion de la scolarité de l'apprenant.

- Mise à la disposition de l'apprenant d'outils de travail (éditeurs spécifiques, outils de téléchargement, outils de simulation de T.P,...).
- Mise à la disposition des différents acteurs d'outils de communications ainsi que les procédés de leurs gestions (forums, messagerie, chat, visioconférence,...).

Il existe un grand nombre de plates-formes d'enseignement à distance sur le marché international, environ plus de 200 dont une trentaine sous licences libres. Parmi les plates-formes sous licence libre (ou licence GPL), nous pouvons citer : Moodle, Claroline, Ganesha.

Il existe aussi des plates-formes sous licence propriétaire comme : e-doceo, myTeacher, Blackboard (nouveau nom depuis 2006 de WEBCT).

II.3. Trace

On appelle trace une collection d'observés temporellement situés. Par le terme observé, nous dénotons toute information structurée issue de l'observation d'une interaction, et par séquence temporelle, nous dénotons l'existence d'une relation d'ordre organisant ces observés par rapport à une extension temporelle qui peut être [4]:

- Soit un intervalle temporel déterminé par deux dates, appelées « date de début » et « fin de l'observation » ;
- Soit une séquence d'éléments quelconques fournissant la relation d'ordre temporelle (par exemple l'ensemble des entiers naturels)

Chaque trace possède sa propre structure et sémantique, et ceci suivant le système qui l'a généré et la nature de l'activité tracée. Donc, pour pouvoir être utilisée, une trace est toujours accompagnée de son modèle. On parle ici de modèle de trace.

II.4. Modèle de traces

On définit le modèle de trace comme le vocabulaire de la trace qui décrit de manière abstraite les objets qui en font partie [4]. D'un point de vue purement informatique, on peut dire que le modèle de trace est un ensemble de classes (classe en sens paradigme orienté objet) et éventuellement leurs relations. Les classes permettent de donner une description aux observés de la trace.

Un modèle de trace peut être implicite, de sorte qu'il est implicitement déduit des données de la trace, comme par exemple le « Common Log Format » pour les traces du

serveur « Apache ». Il peut aussi être explicite, c'est-à-dire définie séparément des données de la trace pour permettre leurs échanges et leurs réutilisations. Différents formats peuvent être utilisés pour représenter les modèles de traces. Dans notre travail et comme on le verra plus loin, le modèle de la trace est une ontologie OWL.

II.5. Trace modélisée

On appelle trace modélisée l'association d'une collection d'observés temporellement situés et d'un modèle explicite de cette collection d'observés. Techniquement, les traces modélisées sont représentées sous forme d'une ontologie OWL [5].

II.6. Indicateur

Un indicateur est une variable au sens mathématique à laquelle est attribuée une série de caractéristiques. C'est une variable qui prend des valeurs de forme numérique, alphanumérique ou même graphique.... La valeur possède un statut : elle peut être brute (sans unité définie), calibrée ou interprétée. Le statut identifie une caractéristique bien précise : celle du type d'assistance offert aux utilisateurs. Chaque indicateur peut dépendre d'autres variables comme le temps, ou même d'autres indicateurs. [6]

III. Quelques travaux sur les E.I.A.H. et les traces

Nous présentons dans cette partie quelques travaux reliant les traces aux E.I.A.H. Nous citons en particulier des travaux sur l'interprétation et la visualisation des traces, ensuite des travaux sur des expérimentations pour recueillir les traces afin de les analyser.

III.1. Visualisations des traces

III.1.1. E-Médiathèque

Ce travail s'intéresse à la visualisation des traces issues des plateformes de téléformation collaboratives. Il est issu du projet eLycée où la trace première est transformée en trace modélisée pour une activité de coproduction dans une dyade. La visualisation des traces est précédée par un traitement statique de la trace première, suivie par un traitement dynamique de la trace métier. La trace est modélisée à travers un modèle d'utilisation. Une fois la trace définie, on applique deux transformations:

- Une transformation statique faite par le système et permettant le passage de la trace première vers la trace métier. Cette transformation est basée sur les transformations disponibles dans le système de gestion de base de traces (transformations conceptuelles).
- Une transformation dynamique faite par l'utilisateur grâce à l'interface graphique (transformation de présentation). [7]

III.1.2. CourseVis

CourseVis [8] est utilisé pour la visualisation en trois dimensions des traces issues de la plateforme WebCT (Figure I.1). Le but est de faire des visualisations basées sur des mesures statistiques.

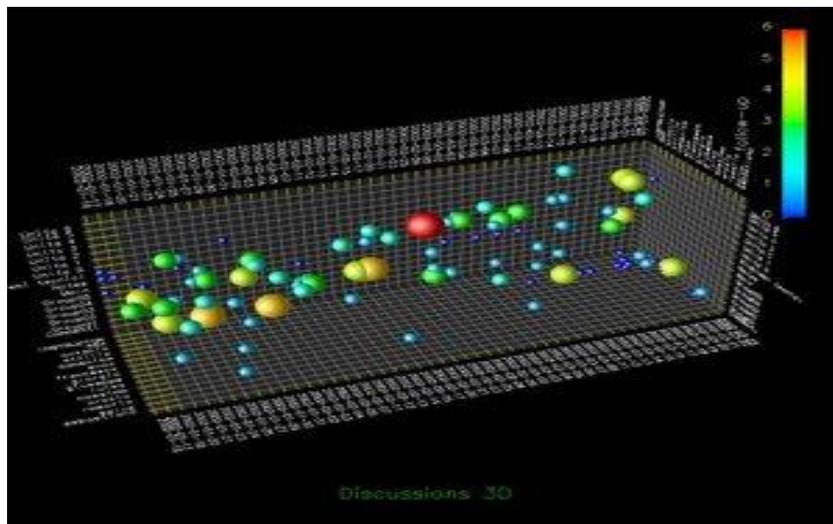


Figure I.1 : CourseVis [8]

La visualisation est précédée par quelques transformations (Figure I.2):

- Sélection des données à partir des sources de traçage. CourseVis utilise le module « Student data exporter » pour importer les données de la plateforme WebCT dans un format XML. Ces données sont stockées dans une Base des données.
- Sélection des données significatives à partir de la Base des données. On ne sélectionne que ce que on veut visualiser, à la demande de l'utilisateur.
- Transformation des données pour les mettre dans un format bien précis.
- Transformation des données sous forme de primitives graphiques qui seront les briques pour la visualisation des données aux utilisateurs.

- Génération de la visualisation graphique à partir des primitives. Cette visualisation peut être paramétrée par les utilisateurs. CourseVis est à destination des tuteurs pour disposer d'une sorte de tableau de bord d'une activité d'apprentissage (suivi d'un groupe).

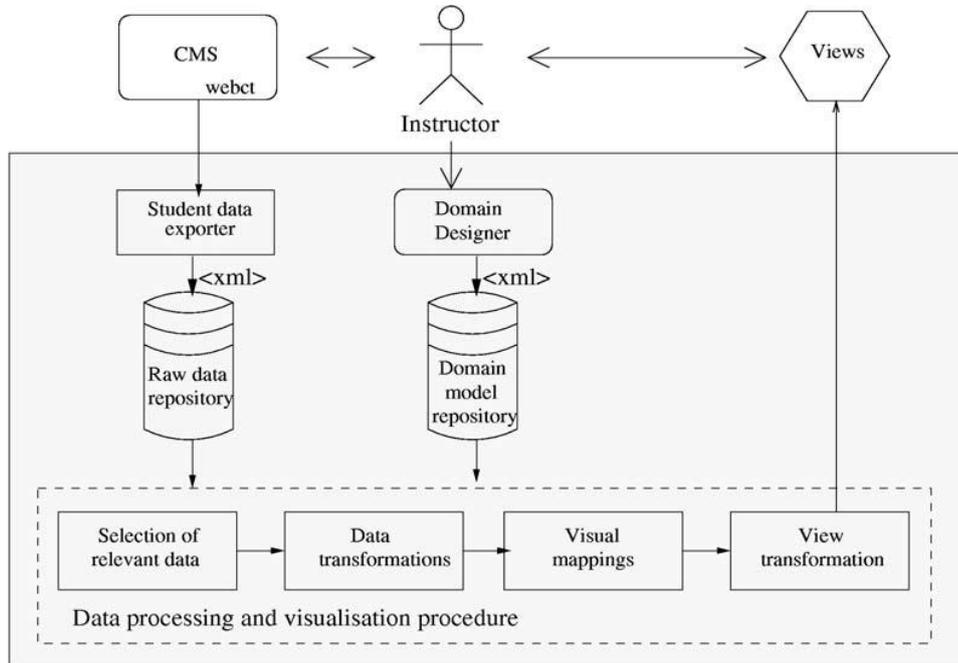


Figure I.2 : Architecture de CourseVis [8]

III.1.3. ClassroomVis

Dans ClassroomVis [9], on explique l'intérêt de visualiser les traces en temps réel ce qui permet au tuteur d'observer l'activité synchrone d'un groupe d'apprenants et d'adapter cette l'activité en interagissant directement avec l'interface de visualisation.

Dans la Figure I.3, on utilise des bulles pour représenter des activités et des liens pour représenter les parcours entre les activités. Les apprenants sont représentés par des visages.

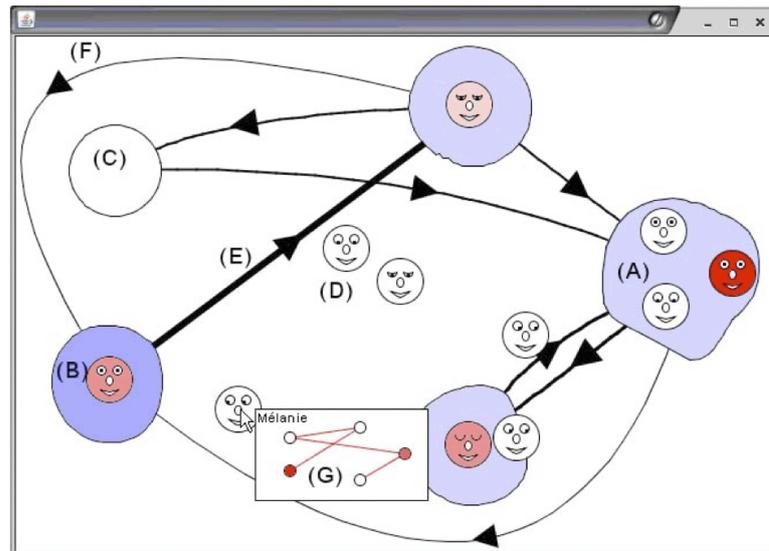


Figure I.3 : ClassroomVis : Une visualisation des états des apprenants, les parcours effectués et les activités réalisées [9]

Ces visualisations évoluent dynamiquement dans le temps, en fonction des traces laissées par l'apprenant dans un intervalle de temps. Par exemple, la forme des yeux du « visage » d'un apprenant (e.g. fermée, ouverte) correspond à la quantité d'événements produits par l'apprenant pour une certaine activité. Si un apprenant « dort » dans une activité particulière, le tuteur peut intervenir en lui suggérant de participer plus activement.

III.2. E.I.A.H. et expérimentations pour recueillir les traces

Afin de délimiter les buts propres à ce mémoire et le justifier dans un sens, il est intéressant de chercher à occasionner le bon compromis en ayant recours à des expérimentations référencées pour la cueillette des traces et la sollicitation de la plateforme Moodle.

Le cas suivant concerne les travaux à Romero, Ventura et Garcia [10] ; ces derniers avaient pour but d'occasionner un apprentissage en sollicitant des points théoriques et pratiques concernant des extractions (automatisées) de données qui se révéleront essentielles pour des buts de visualisations et de classifications.

La figure suivante explicitera le processus des démarches faites pour les classifications en question.

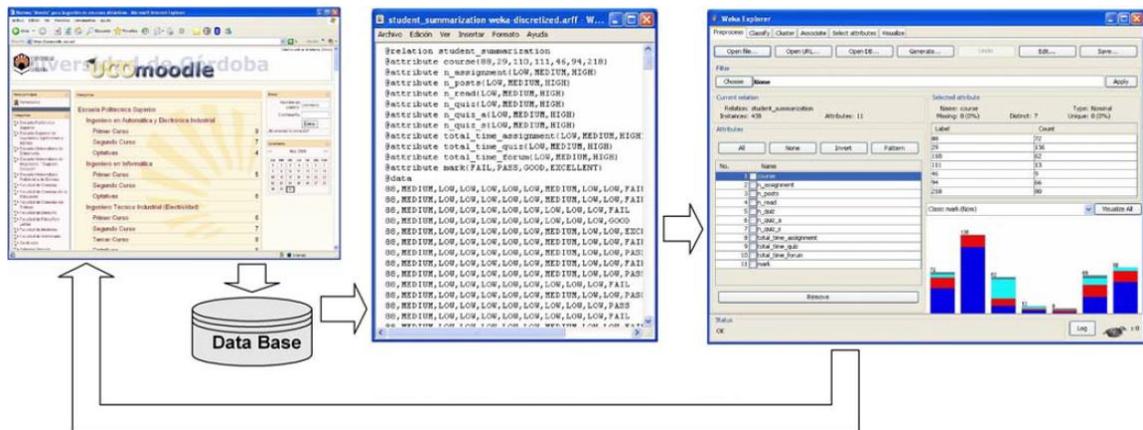


Figure I.4 : Extraction des données Moodle [10]

Comme il est visible sur la figure, le processus se déroule comme suit :

- Collecter des données (générales) à partir de la plateforme Moodle.
- Les données collectées sont remaniées et transformées pour une éventuelle classification.
- Des algorithmes sont appliqués pour construire et déployer le modèle qui jouera un rôle pour délimiter les connaissances propres aux intérêts de l'utilisateur (Enseignant, étudiant, administrateur...).
- Les résultats seront interprétés, évalués et déployés. L'enseignant pourrait à titre d'exemple utiliser les informations générées pour effectuer des décisions sur les apprenants au sein de la plateforme Moodle, dans un but d'apprentissage et/ou d'amélioration.

Un autre cas intéressant concerne celui de T.DJAOUAD [11] qui dans son travail a cherché à synthétiser des « indicateurs », ces indicateurs se matérialiseront par ce qui suit :

- Intégration d'un système à base de traces comme nouvelle brique disponible pour la conception d'E.I.A.H. (intégration dans un Framework de type Moodle).
- Elaboration d'un processus intégré de transformations explicites de traces modélisées préparant à un calcul explicite d'indicateurs.

Les connaissances de modélisation des traces, des transformations et de la description de la formule d'indicateur sont capitalisées, facilitant la réutilisation et la constitution de bibliothèques d'indicateurs « prêts à l'emploi ».

L'outil développé pour faire la démonstration de l'approche proposée « Moodle » a été testé en vraie grandeur sur plusieurs terrains universitaires.

La figure suivante montre l'architecture faite pour cette thèse :

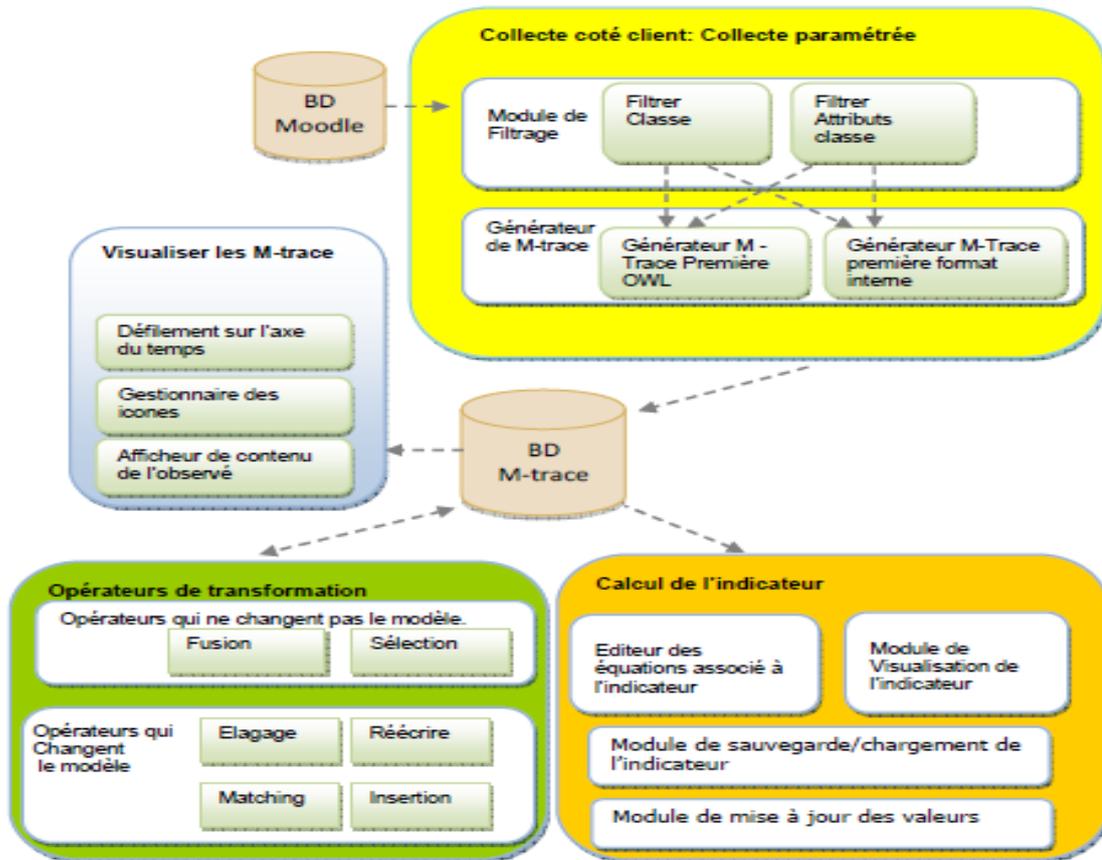


Figure I.5 : Architecture du système [11]

La figure présente l'œuvre de T.DJAOUAD qu'est le « SBT-IM »: Système à Base de Traces pour calculer les indicateurs dans Moodle. Le SBT-IM comprend quatre grandes parties:

- Un sous-système de collecte des données à partir des plateformes d'apprentissages (Moodle).
- Un sous-système qui transforme les M-traces.
- Un sous-système de visualisation des M-traces.
- Un sous-système qui calcule les indicateurs à l'issue d'un processus de transformation des M-traces.

III.3. Synthèse

On présentera lors de cette synthèse un tableau récapitulatif (tableau I.1) des différents travaux que nous avons cité sur la visualisation de traces. Pour chaque travail on identifiera comment il répond aux besoins «Pratiques» que nous cherchons à satisfaire par notre système.

Les besoins de notre système	E-Mediatheque	CourseVis	ClassroomVis
Sélectionner les sources de traçage	Source unique Fichiers logs de la plateforme E-Mediatheque	Source Unique Données issue des fichiers logs	Source Multiple Données issue de plusieurs fichiers logs
Collecter des observations à partir des sources de traçage	Intégrée dans la plateforme d'apprentissage E-Mediatheque	Indépendamment de la plateforme d'apprentissage	Intégrée dans l'outil, Basée sur des agents
Préparer les données transformées pour le calcul d'indicateur	Une seule transformation entre la trace première et la trace métier	Un prétraitement basé sur des mesures statistiques	Prétraitement intégré fait par un agent informatique
Visualisation accessible par	Tuteur/apprenant selon les droits	Tuteur	-
Associer les traces aux activités de l'indicateur	Non	Non	Non

Tableau I.1 : Synthèse sur les différents travaux de visualisation de la trace, et identification de nos besoins par rapport à ces travaux

La comparaison entre ces différents travaux nous permet de pouvoir matérialiser nos besoins dans un but de visualisation des traces d'interaction. Nous sélectionnons nos besoins de la façon suivante :

- Identifier les compromis entre les différents systèmes de visualisation de trace, puis extraire ces compromis et les incorporer dans notre futur système.
- Identifier les lacunes des systèmes et les incorporer dans notre futur système.

Il est à rappeler que le mémoire ne s'intéresse pas à la visualisation des traces, néanmoins, il est nécessaire de la faire. La visualisation de la trace elle-même est un moyen qui aide à l'élaboration des indicateurs à partir du contenu de la trace en question.

Les travaux de ROMERO & al. [10] s'intéressent quant à eux au processus de l'extraction des données à partir de la plateforme « Moodle ». Par rapport aux travaux

de T.DJAOUAD, notre travail s'inspire des méthodes de modélisations de traces proposées, et l'élaboration des indicateurs par des règles de calculs. L'analyse de ces travaux va nous permettre d'étendre les possibilités offertes sur la plateforme « Moodle » en ajoutant une chose nouvelle qui est la « génération de profils ».

IV. Reformulation de la problématique

A l'instar des tuteurs intelligents, les plateformes d'apprentissages telles que « Moodle » sont pauvres en matière d'interaction avec l'apprenant dans une session d'apprentissage.

L'utilisation des traces peut en effet palier à ce manque car leurs exploitations peuvent nous aider à estimer, calculer et cerner le profil et faire le suivi des apprenants. Nous avons vu dans ce chapitre à travers l'ensemble des travaux cités précédemment que l'utilisation des traces est un moyen efficace pour améliorer et aider le diagnostic des différentes composantes (apprenant, cours,...) d'un système d'apprentissage.

L'utilisation de ces systèmes d'apprentissages est très répandue dans différentes universités en particulier la plateforme « Moodle », d'ailleurs notre université est dotée d'une plateforme d'apprentissage « elearn.univ-tlemcen.dz » qui repose sur « Moodle ».

Le problème qui se pose est que « Moodle » donne juste des statistiques concernant les apprenants par rapport à un cours, un ensemble d'enseignants préférerait en plus des statistiques, avoir des indices, avoir des indicateurs qui les renseigneront sur l'état des apprenants vu que l'apprentissage se fait à distance, de ce fait : Comment profiter et exploiter les traces de « Moodle » pour arriver à faire une première analyse de l'état de l'apprentissage des étudiants pour générer leurs profils ?

Cette génération de profil de l'étudiant, suivant bien sûr des paramètres établis par l'enseignant, peut l'aider à rapidement prendre connaissance de l'état d'apprentissage et du comportement de ses apprenants par rapport à son cours. Dans le reste du mémoire nous montrerons à travers le modèle « TrMood » : Modèle de trace de « Moodle » et « GenProApp » : système de Génération de profils d'apprenant, les étapes suivies pour l'analyse, la modélisation et l'exploitation des traces de « Moodle » pour les fins de génération du profil d'apprenant.

Chapitre II
Modélisation des traces
brutes de Moodle

I. Introduction

Dans ce chapitre, nous présentons la démarche de collecte des traces à partir de la plateforme « Moodle », nous détaillons les tables utilisées pour cette collecte, nous présentons ensuite le modèle de la trace première « l'ontologie TrMood » (les traces résultantes du processus de la collecte, sont appelées « traces premières ») conçue pour Moodle ainsi que le traitement des traces modélisées avec le calcul d'indicateurs à travers des règles pour la génération du profil qui suivront une échelle de notation choisie.

II. Analyse des données de Moodle (traces)

Dans notre travail nous avons eu recours à la plateforme d'apprentissage en ligne « Moodle »¹, la raison de ce choix est dû au fait qu'elle soit utilisée par notre université et qu'elle soit sous licence libre (contrainte d'accessibilité); de par ce fait, « Moodle » est une des plateformes des plus répandues de par son recours d'un grand nombre d'utilisateurs nonobstant une implantation dans plus de 50 000 sites et démocratisé dans plus de 208 pays.

Contrairement à d'autres plateformes, « Moodle » ne stocke pas les données sous forme de fichiers logs (*.logs), mais plutôt dans une base de données relationnelle. Ainsi, le stockage est centralisé dans une seule base de données.

Il est intéressant de savoir que le nombre de tables dans la base de données de Moodle varie selon la version, à titre d'exemple, dans la version 2.4.2 ; Il existe en total 305 tables « Mysql »². Nous nous sommes contentés de quelques tables (qui se veulent essentielles) et qui ont servi pour la collecte des données. Chaque table contient plusieurs champs. Nous n'avons aussi illustrés que les champs sollicités dans la collecte.

- La table User

mdl_user		
Nom de champs	Type	Description du champ
Id	bigint(10)	Identifiant de l'utilisateur
username	varchar(100)	Pseudonyme de l'utilisateur
firstname	varchar(100)	Le nom de l'utilisateur

Tableau II.1 : La table mdl_user

¹ Moodle : Modular Object Oriented Dynamic Learning Environment .

² MySQL : Est un système de gestion de base de données (SGBD).

Chapitre II : Modélisation des traces brutes de Moodle

- La table Cour

mdl_course		
Nom de champs	Type	Description du champ
Id	bigint(10)	Identifiant du cours
shortname	varchar(255)	Le nom abrégé du cours

Tableau II.2 : La table mdl_course

- La table Log

mdl_log		
Nom de champs	Type	Description du champ
Id	bigint(10)	Identifiant de l'action
Time	bigint(10)	L'instant de l'action.
Userid	bigint(10)	Identifiant de l'utilisateur
Course	bigint(10)	Identifiant du cours
Module	varchar(20)	L'activité en question
Action	varchar(40)	Eventuelle action effectuée par l'apprenant dans un module.
Info	varchar(255)	Information concernant l'action

Tableau II.3 : La table mdl_log

- Les tables de Lesson

mdl_lesson		
Nom de champs	Type	Description du champ
Id	bigint(10)	Identifiant de la leçon
Name	varchar(255)	Le nom de la leçon

Tableau II.4 : La table mdl_lesson

mdl_lesson_pages		
Nom de champs	Type	Description du champ
Id	bigint(10)	Identifiant du cours
Lessonid	bigint(10)	Le nom de la leçon

Tableau II.5 : La table mdl_lesson_pages

mdl_lesson_timer		
Nom de champs	Type	Description du champ
Id	bigint(10)	La clé primaire de la table
Lessonid	bigint(10)	Le nom abrégé du cours
Userid	bigint(10)	Identifiant de l'utilisateur
starttime	bigint(10)	Date de début de la leçon
lesstime	bigint(10)	Date de fin de la leçon

Tableau II.6 : La table mdl_lesson_timer

mdl_lesson_grades		
Nom de champs	Type	Description du champ
Lessonid	bigint(10)	Identifiant de la leçon
Userid	bigint(10)	Identifiant de l'utilisateur
Grade	Double	La note du test dans la leçon

Tableau II.7 : La table mdl_lesson_grades

- Les tables du Quizz

mdl_quizz		
Nom de champs	Type	Description du champ
Id	bigint(10)	Identifiant du quizz
Name	varchar(255)	Le nom du quizz

Tableau II.8 : La table mdl_quizz

mdl_quizz_attempts		
Nom de champs	Type	Description du champ
Id	bigint(10)	La clé primaire de la table
Quizz	bigint(10)	Identifiant du quizz
Userid	bigint(10)	Identifiant de l'utilisateur
timestart	bigint(10)	Date de début du quizz
timefinish	bigint(10)	Date de fin du quizz

Tableau II.9 : La table mdl_quizz_attempts

mdl_quizz_grades		
Nom de champs	Type	Description du champ
Quizz	bigint(10)	Identifiant du quizz
Userid	bigint(10)	Identifiant de l'utilisateur
Grade	decimal(10,5)	La note du quizz

Tableau II.10 : La table mdl_quizz_grades

- Les tables de Chat

mdl_chat		
Nom de champs	Type	Description du champ
Id	bigint(10)	Identifiant chat
Course	bigint(10)	Identifiant du cour
Name	varchar(255)	Le nom du chat

Tableau II.11 : La table mdl_chat

mdl_chat_messages		
Nom de champs	Type	Description du champ
Id	bigint(10)	Identifiant message
Chatid	bigint(10)	Identifiant du chat
Userid	bigint(10)	Identifiant utilisateur
Message	Longtext	Le message envoyé dans le chat
timestamp	bigint(10)	Le temps associé au message du chat

Tableau II.12 : La table mdl_chat_messages

- Les tables de Forum

mdl_forum		
Nom de champs	Type	Description du champ
Id	bigint(10)	Identifiant forum
Name	varchar(255)	Le nom de forum

Tableau II.13 : La table mdl_forum

mdl_forum_discussion		
Nom de champs	Type	Description du champ
Id	bigint(10)	Identifiant de la discussion
Forum	bigint(10)	Le nom de forum

Tableau II.14 : La table mdl_forum_discussions

mdl_forum_posts		
Nom de champs	Type	Description du champ
discussion	bigint(10)	Identifiant de la discussion
Message	Longtext	Le message écrit dans le forum

Tableau II.15 : La table mdl_forum_posts

Après avoir présenté les tables que nous avons utilisées pour la collecte, nous allons définir le modèle de trace première et le transformer en une ontologie³ « OWL »⁴.

III. Modélisation des traces

III.1. Modèle de trace première

À l'issue de la première analyse, nous avons construit une trace première à partir des tables de « Moodle ». Pour concevoir le modèle de la trace première nous nous sommes basé sur le travail de T.Djaouad [12], nous avons amélioré ce modèle pour les activités collaboratives utilisant les ressources précédemment citées.

Il a été étendu pour prendre en compte les actions supplémentaires concernant la leçon et le quizz. Le modèle respecte naturellement un méta-modèle que toutes les traces respecteront. Nous distinguons dans ce méta-modèle quatre classes :

- La classe « observed » qui représente l'action qu'on souhaite observer. Par exemple : voir un cours, écrire dans un forum, répondre à un quiz, etc...
- La classe « user » associé à l'observé. C'est l'utilisateur qui fait l'action.
- La classe « times » associé à l'observé. C'est le temps de l'action.
- La classe « tools » associé à l'observé. C'est l'outil qui supporte l'interaction.

La Figure II.1 illustre ce méta-modèle en format UML⁵.

³ Ontologie : une ontologie est une représentation de connaissances (ingénierie des connaissances).

⁴ OWL : est un vocabulaire XML basé sur RDF pour définir des ontologies

Chapitre II : Modélisation des traces brutes de Moodle

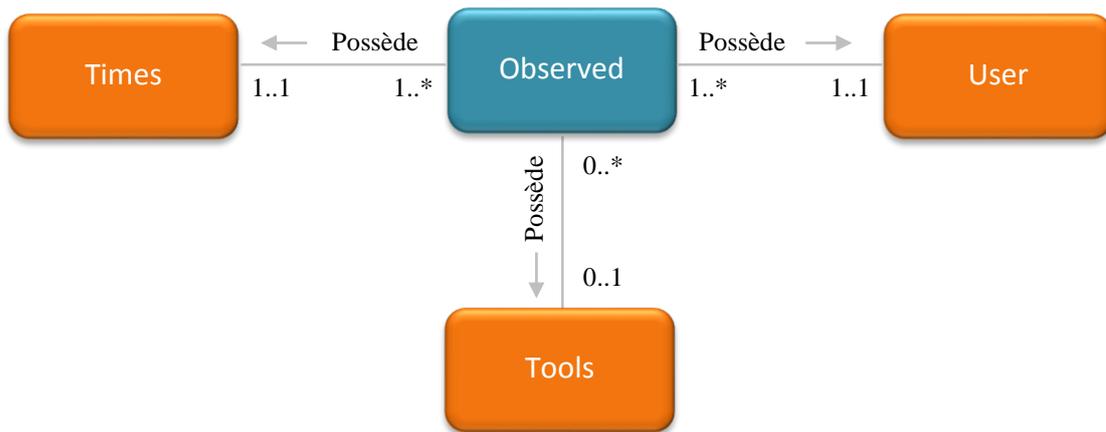


Figure II.1 : Le méta-modèle de la trace première en format UML

Dans ce méta-modèle nous proposons les règles de cardinalités suivantes :

- Un observed est associé à un et un seul user.
- Un user peut avoir un ou plusieurs observed.
- Un observed est associé à un et un seul tools.
- Tools peut avoir un ou plusieurs observed.
- Un observed est associé à une et une seule valeur de times.
- Une valeur de times peut avoir un ou plusieurs observed.

Nous avons par la suite instancié ce méta-modèle. Pour les deux classes « tools » et « Observed ». Moodle propose plusieurs outils: Chat, Forum, Glossaire, Wiki, Sondage, Test, etc. Nous avons sélectionné les quatre outils : Chat, Forum, Lesson, et Quizz. Nous avons par la suite identifié les observés que nous souhaitons tracer.

Par outil, voici les observés que nous modélisons :

- Cours :
 - Voir (consulter) un cours
- Leçon :
 - Voir (consulter) une leçon
 - Répondre à des questions dans la leçon
- Quizz :
 - Voir (consulter) un quizz
 - Répondre à un quizz

⁵ UML : sigle désignant l'Unified Modeling Language ou « langage de modélisation unifié » est un langage de modélisation graphique

- Chat :
 - Ecrire un message dans un chat
 - Entrer dans un chat
- Forum :
 - Envoyer un message dans un forum
 - Créer un fil de discussion dans un forum
 - Voir (consulter) un forum

La Figure II.2 représente le modèle de la trace première « Moodle » en format UML.

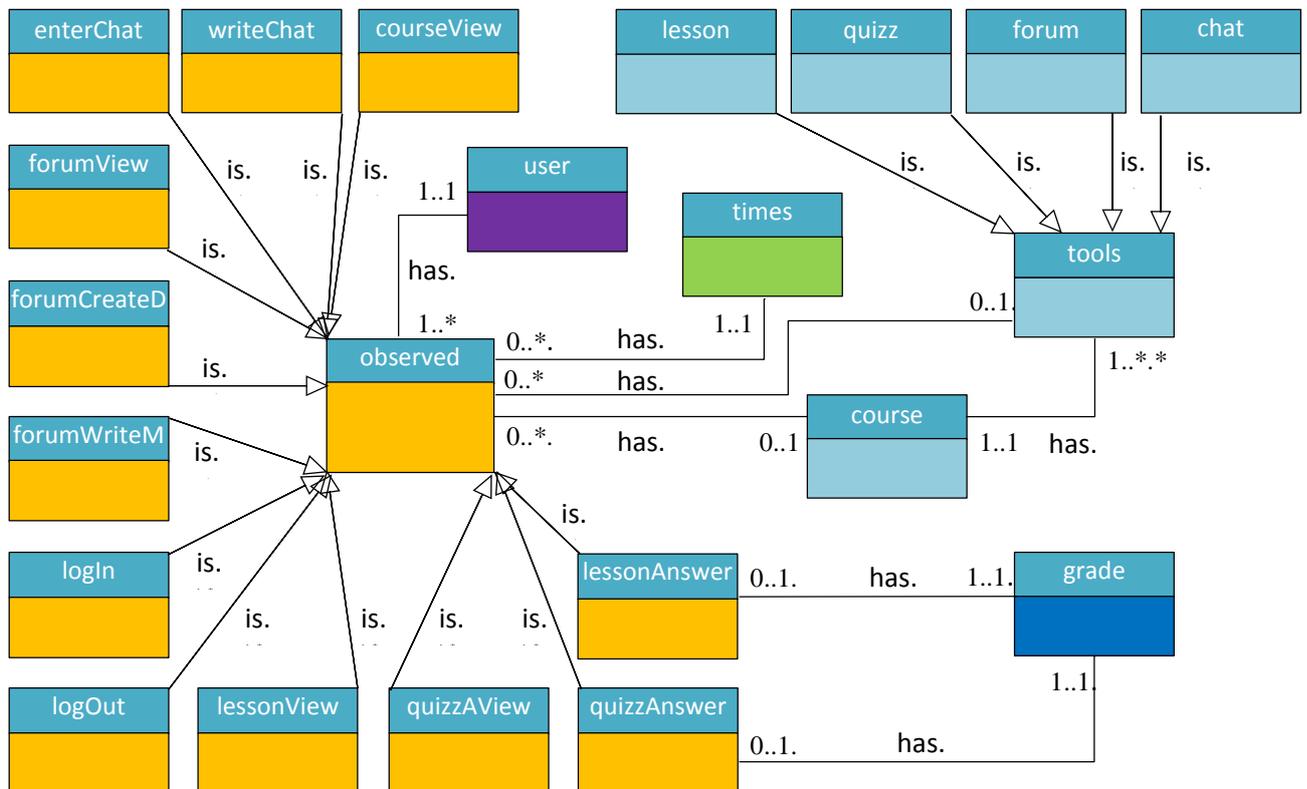


Figure II.2 : Le méta-modèle

Cette première mise en œuvre a permis de modéliser l'observation d'une activité d'apprentissage (modèle de trace première) accommodé par une plateforme « Moodle », à partir des logs existants, sans instrumentation particulière nécessaire pour l'enseignant ou le tuteur.

III.2. L'ontologie TrMood

Afin de faire nos analyses sur la trace première, nous avons transformé le modèle proposé précédemment en une ontologie OWL, elle est basée sur six entités : les observés, l'acteur, le cours, le temps, la note et les outils. Les observés représentent

Chapitre II : Modélisation des traces brutes de Moodle

tout ce qu'on souhaite observer comme actions faites par les acteurs. Ces observés sont liés à l'utilisation des différents outils offerts par « Moodle » (chat, forum, lesson et quizz), nous avons formalisé cette ontologie avec l'outil Protégé⁶.

La figure II.3 présente une copie d'écran du modèle conceptuel décrit sous Protégé.

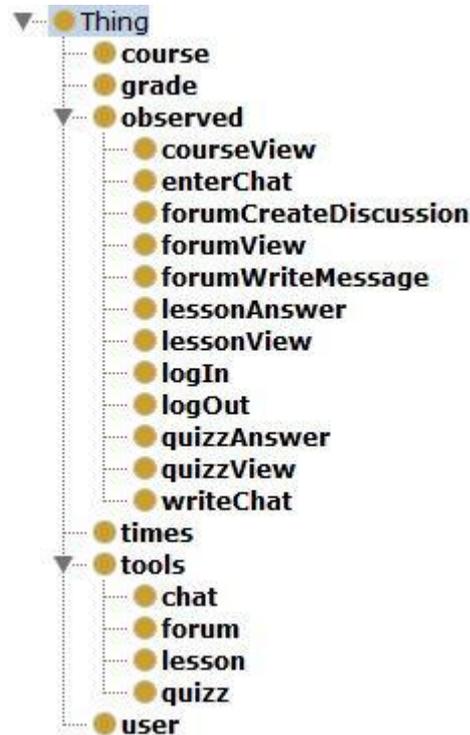


Figure II.3 : Capture d'écran de Protégé : concepts du modèle d'utilisation correspondant à la trace première

Notre ontologie est composée de 6 classes : course, grade, observed, times, tools et user. La classe observed est composée des sous-classes suivantes : courseView, enterChat, writeChat, forumCreateDiscussion, forumView, forumWriteMessage, lessonAnswer, lessonView, logIn, logOut, quizzView et quizzAnswer.

La classe tools est composée des quatre sous-classes : lesson, quizz, chat et forum.

Nous avons défini six propriétés d'objets dans notre ontologie : hasCourse, hasUser, hasTool, hasGrade, hasTime et hasDo.

La figure II.4 présente une copie d'écran de ces propriétés décrit sous « Protégé ».

⁶ Protégé : est un éditeur d'ontologie gratuit et open source développé en java.

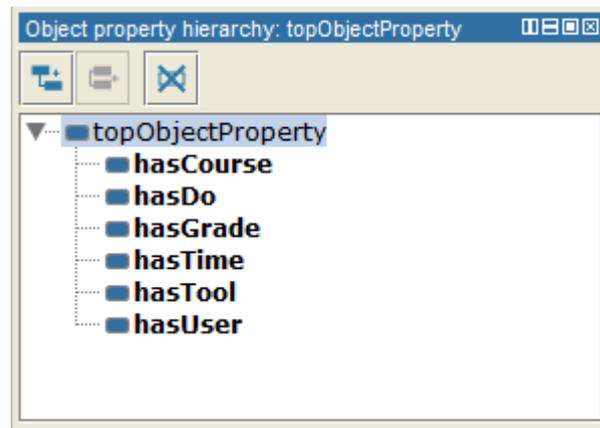


Figure II.4 : Copie écran Protégé : propriété d'objet du modèle d'utilisation

- hasCourse : cette relation relie la classe observed avec la classe course.
- hasUser : cette relation relie la classe observed avec la classe user.
- hasTool : cette relation relie les classes course et observed avec la classe tools.
- hasGrade : cette relation relie les classes quizzAnswer, lessonAnswer et course avec la classe grade.
- hasTime : cette relation relie la classe observed avec la classe times.
- hasDo : est une propriété inverse de la propriété hasUser.

Nous avons aussi définie quatre propriétés de données : hasName, hasNote, inTime et textWritten. La figure II.5 présente une copie d'écran de ces propriétés décrit sous Protégé.

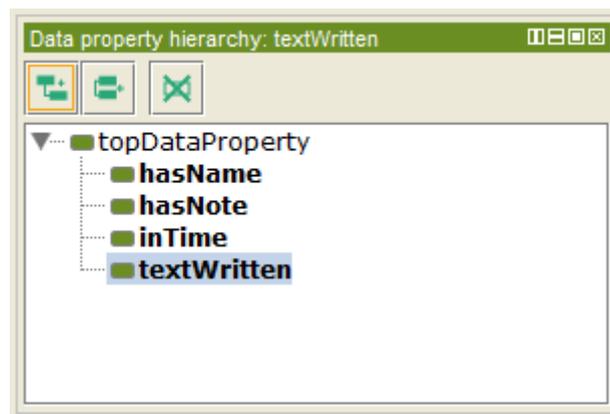


Figure II.5 : Copie écran Protégé : propriété de données du modèle d'utilisation

- hasName : elle est reliée avec les classes tools et user, elle est de type String et de cardinalité 1.
- hasNote : elle est de type String et elle est reliée avec la classe grade.
- inTime : elle est de type String et elle est reliée avec la classe times.

Chapitre II : Modélisation des traces brutes de Moodle

- `textWritten` : elle est reliée avec les classes `forumWriteMessage` et `writeChat`, elle est de type `String`.

La figure II.6 représente une copie d'écran de notre ontologie bien détaillée avec les classes, les sous-classes, les propriétés d'objets ainsi que les propriétés de données.

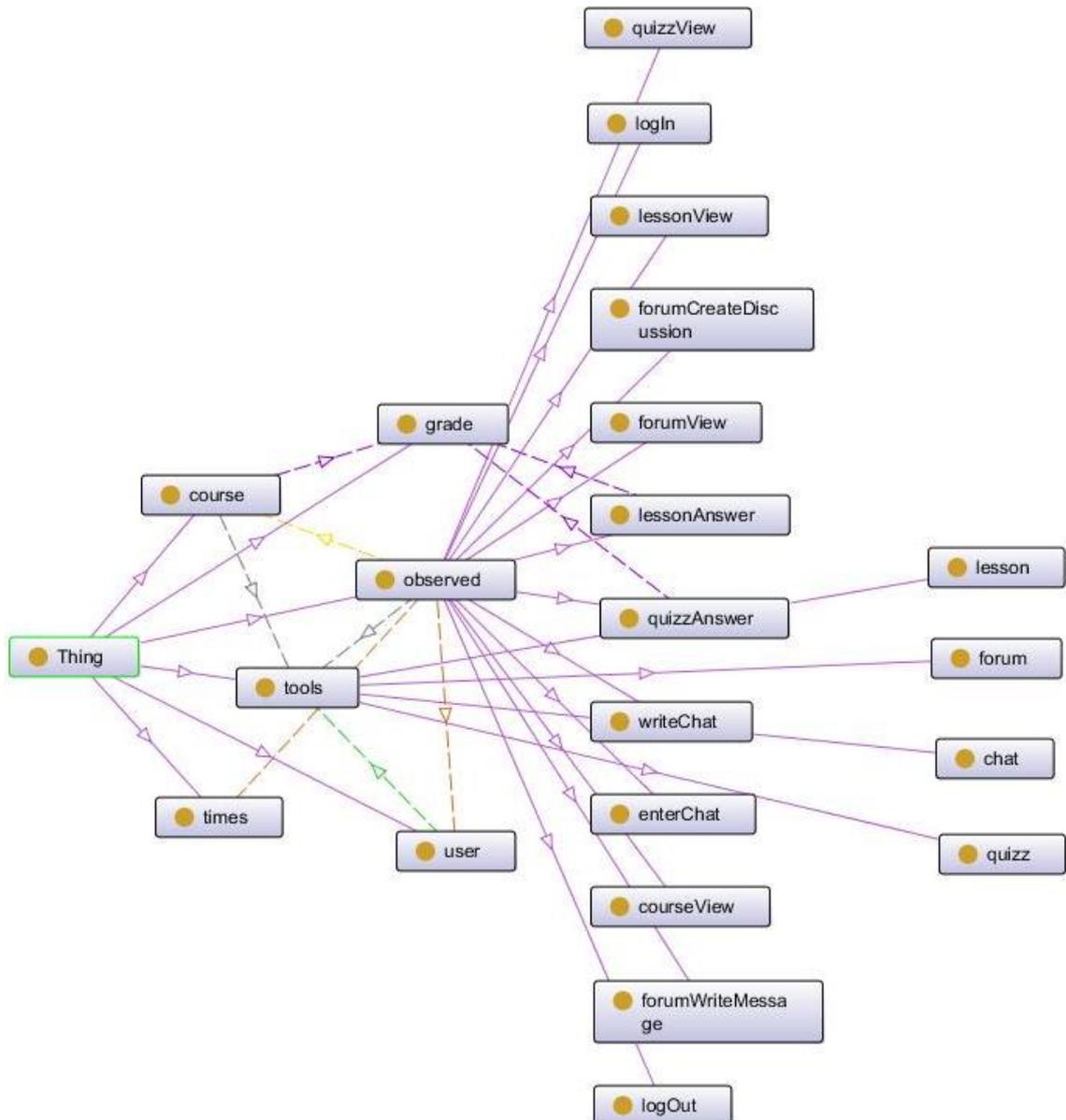


Figure II.6 : Copie écran Protégé : L'ontologie OWL visualisée avec « OntoGraf »⁷

Le modèle d'utilisation ainsi défini, nous avons généré automatiquement toutes les instances du modèle d'utilisation à partir de la source de traçage « Moodle », la figure II.7 montre un observé qui représente une action de type « répondre à un Quizz » effectuée par l'utilisateur répondant au pseudonyme 'sofyane46'.

⁷ OntoGraf : un plugin associé à Protégé qui permet d'afficher la hiérarchie des classes avec les relations entre eux.

```
<quizzAnswer rdf:about="http://www.semanticweb.org/zenati/ontologies/2013/3/untitled-ontology-8#idQuizzAnswer1">
  <hasUser rdf:resource="http://www.semanticweb.org/zenati/ontologies/2013/3/untitled-ontology-8#sofyane46"/>
  <hasTool rdf:resource="http://www.semanticweb.org/zenati/ontologies/2013/3/untitled-ontology-8#idQuizz1"/>
  <hasTime>
    <times rdf:about="http://www.semanticweb.org/zenati/ontologies/2013/3/untitled-ontology-8#idTimeQuiz1">
      <inTime>591</inTime>
    </times>
  </hasTime>
  <hasGrade>
    <grade rdf:about="http://www.semanticweb.org/zenati/ontologies/2013/3/untitled-ontology-8#idQuizzGrade1">
      <hasNote>11.92</hasNote>
    </grade>
  </hasGrade>
  <hasCourse rdf:resource="http://www.semanticweb.org/zenati/ontologies/2013/3/untitled-ontology-8#BDD"/>
</quizzAnswer>
```

Figure II.7 : Un exemple d'observé écrit en OWL, généré par notre application

IV. Formalisation des règles de calculs des indicateurs

Nous expliquons dans cette partie comment nous avons élaboré et calculé les indicateurs, nous donnons pour chaque indicateur que nous calculons une description détaillée, sa règle de calcul et les valeurs générés.

Nous choisissons de calculer les indicateurs qui se basent sur le temps passé dans le cours, le temps passé pour répondre aux questions de « lesson » et de « quizz » ainsi que la participation dans le « forum ».

IV.1. Calcul d'indicateurs

Pour générer le profil d'apprenant on s'est basé sur une note, cette note est calculée à partir de quatre indicateurs : un indicateur de « quizz », un indicateur de « cours » ainsi qu'un autre de « leçon » et un dernier pour « forum », chaque indicateur est associé à une règle de calcul mathématique.

A partir des traces modélisées : On extrait pour chaque calcul d'indicateur les valeurs occasionnées par l'apprenant lors de son examination des modules « quizz » et « lesson » via une requête SPARQL⁸.

Les calculs d'indicateurs ont des particularités, afin de les dissocier, on les citera comme étant des « modes », on en dénombre deux, pour lesquels on a attribué des noms respectivement: « par défaut » et « avancé ».

⁸ SPARQL : est un langage de requête qui permet de manipuler des données RDF.

IV.1.1. Mode « par défaut »

Le calcul s'effectue comme suit :

- L'indicateur « quizz » \Rightarrow Retourne la note de « quizz ».
- L'indicateur « lesson » \Rightarrow Retourne la note de « lesson ».
- Le score global de l'apprenant *Noteg* \Rightarrow Calcule l'opération suivante :

$$Noteg = \frac{Note \text{ « quizz »} * 2 + Note \text{ « lesson »}}{3}$$

IV.1.2. Mode « avancé »

Le mode « avancé » ne se contente pas d'une simple multiplication pour déterminer les indicateurs comme il a été fait dans le mode « par défaut », en effet ; l'enseignant saisit en premier lieu et selon ses préférences la valeur d'une durée optimale (ou idéale) ainsi qu'un coefficient.

➤ L'indicateur « quizz »

On définit le paramètre « *TempsOptQ* » comme étant le temps optimal saisi initialement par l'enseignant. « *TempsPasséQ* » comme étant la durée concrète quant à la durée passée par l'apprenant dans un module « Quiz », « *NoteQuiz* » désigne la note obtenue de l'apprenant dont la limite (dans cette application) est la valeur 20.

$$IQ = NoteQuiz * \sqrt[4]{\sin\left(\left(\frac{TempsOptQ}{TempsPasséQ}\right) * 90\right)}$$

➤ L'indicateur « lesson »

On définit le paramètre « *TempsOptL* » comme étant le temps optimal opté initialement par l'enseignant. « *TempsPasséL* » comme étant la durée concrète quant à la durée passée par l'apprenant dans un module « Lesson », « *NoteLesson* » désigne la note obtenue de l'apprenant dont la limite (dans cette application) est la valeur 20.

$$IL = NoteLesson * \sqrt[4]{\sin\left(\left(\frac{TempsOptL}{TempsPasséL}\right) * 90\right)}$$

➤ L'indicateur « course »

On définit le paramètre « *TempsOptC* » comme étant le temps optimal entré par l'enseignant. « *TempsPasséC* » comme étant la durée concrète passée par l'apprenant dans un module « *cours* », « *NoteCours* » désigne la note maximale fixée par l'enseignant.

$$IC = NoteCours * \sqrt[4]{\sin\left(\left(\frac{TempsOptC}{TempsPasséC}\right) * 90\right)}$$

Remarque(s)

- L'opération $\sqrt[4]{\sin\left(\left(\frac{tempsOptL}{tempsPasséL}\right) * 90\right)}$ vise « la normalisation », qui se justifie et se base sur ce qui suit :
 - Si « *TempsPassé* » tend vers l'infini (∞) alors le ratio $\left(\left(\frac{TempsOpt}{TempsPassé}\right) * 90\right)$ tend vers le zéro ; par conséquence, le sinus tend aussi vers le zéro ; alors cela diminuera la note du cours.
 - L'utilité de la racine quadratique se traduit par le fait qu'elle allège la sanction que pourrait écoper un apprenant.
 - Etant donné que le temps passé (concret) par un apprenant lors d'une passation d'un module pourrait être :
 - ($\geq TempsOpt$) : la valeur du ratio en question sera inférieure ou égale à la valeur 1, afin de normaliser la valeur obtenue lors du calcul du ratio, on multiplie dans un premier temps cette dernière par le nombre 90 ; puis on introduit le sinus sur l'opération calculée. Le résultat obtenu sera multiplié par la note maximale pour obtenir la note finale de l'apprenant pour un module. (L'apprenant sera dit « normal »)
 - ($\leq TempsOpt$ & $\geq TempsOpt / 2$) : la note finale de l'apprenant pour un module sera maximale. (L'apprenant sera dit « doué »).
 - ($< TempsOpt / 2$) : (L'apprenant sera dit en situation de « gaming »), l'apprenant écoperait d'une sanction dont la variable est « *SancGam* », elle représente le pourcentage paramétré par l'enseignant qui permettra d'infliger ladite sanction . La règle de calcul sera comme suit :

$$\begin{cases} NoteLesson = NoteLesson * SancGam \\ NoteQuizz = NoteQuizz * SancGam \end{cases}$$

➤ L'indicateur « forum »

On définit le paramètre « *NbConsultMoy* » comme étant le nombre de consultation jugé pertinent saisi initialement par l'enseignant. « *NbConsult* » comme étant le nombre de consultation effectif quant à la participation d'un apprenant dans un module « *Forum* », « *NoteForum* » désigne la meilleure note fixée par l'enseignant.

$$IF = NoteForum * \sqrt[4]{\sin\left(\left(\frac{NbConsultMoy}{NbConsult}\right) * 90\right)}$$

Remarque

- La note du module « forum » d'un apprenant se matérialise à partir de l'instant où il accède dans ce module sans pour autant concrètement participer aux activités possibles (tel que créer une discussion ou répondre à une discussion ...)

Ces règles établies, on peut estimer que le processus de génération de profil sera prêt pour une éventuelle expérimentation via l'application faite pour ça, fait qui sera détaillée dans le prochain chapitre.

IV.2. Echelle de notation

Affecter une note à un apprenant varie selon un paramètre géographique, cette contrainte a été prise en considération dans notre système afin d'incorporer un concept d'interprétation d'une notation numérique vers son équivalent sémantique et vice versa, pour ce faire ; il est indispensable de se fier à des normes officielles, notre système prend en charge cinq (05) échelles de notations, la figure suivante explicitera les équivalences des différents systèmes :

Systèmes de notation					
Français	Mention	E.C.T.S.	Américain	Anglais	Suisse
de 16/20 à 20/20	Très bien	A	A à A+ (4.0)	70 ou plus	6
de 15/20 à 16/20	Bien	B	B+ à A-	60-69	5.5
de 13/20 à 14/20	Assez bien	C	B (3.0)	55-59	5
de 11/20 à 12/20	Moyen	D	C+ à B-	50-54	4.5
égale à 10/20	Passable	E	C (2.0)	40-49	4
de 8/20 à 9/20	Oral de rattrapage	A	F	30-39	3.5
inférieur à 8/20	Echec définitif	F	F	moins de 30	3

Tableau II.16 : Les systèmes de notation [13]

V. Conclusion

Dans ce chapitre nous avons montré l'analyse, la collecte et la modélisation des traces « TrMood », nous avons aussi formalisé les règles de calculs pour les différents indicateurs qui seront utilisés par la suite dans notre application enfin on a donné un aperçu sur les notations possibles des apprenants, il est à présent possible d'entamer le prochain chapitre qui va définir le système « GenProApp » qui repose sur le modèle « TrMood » pour générer le profil d'apprenant.

Chapitre III
Génération du profil
d'apprenant

I. Introduction

Ce chapitre s'entamera par des planifications schématiques qui clarifieront le déroulement et la logique qui s'opère au sein du système « GenProApp », il se clôturera par un déroulement d'un scénario d'utilisation du logiciel sur un échantillon (Etudiants en « Master niveau II »), les captures visuelles exposées dans ce chapitre affirmeront ou infirmeront le bon fonctionnement de notre système quant à la génération d'un profil.

La notion de profil dans ce mémoire représente la mention obtenue selon la note de chaque apprenant.

II. Architecture conceptuelle

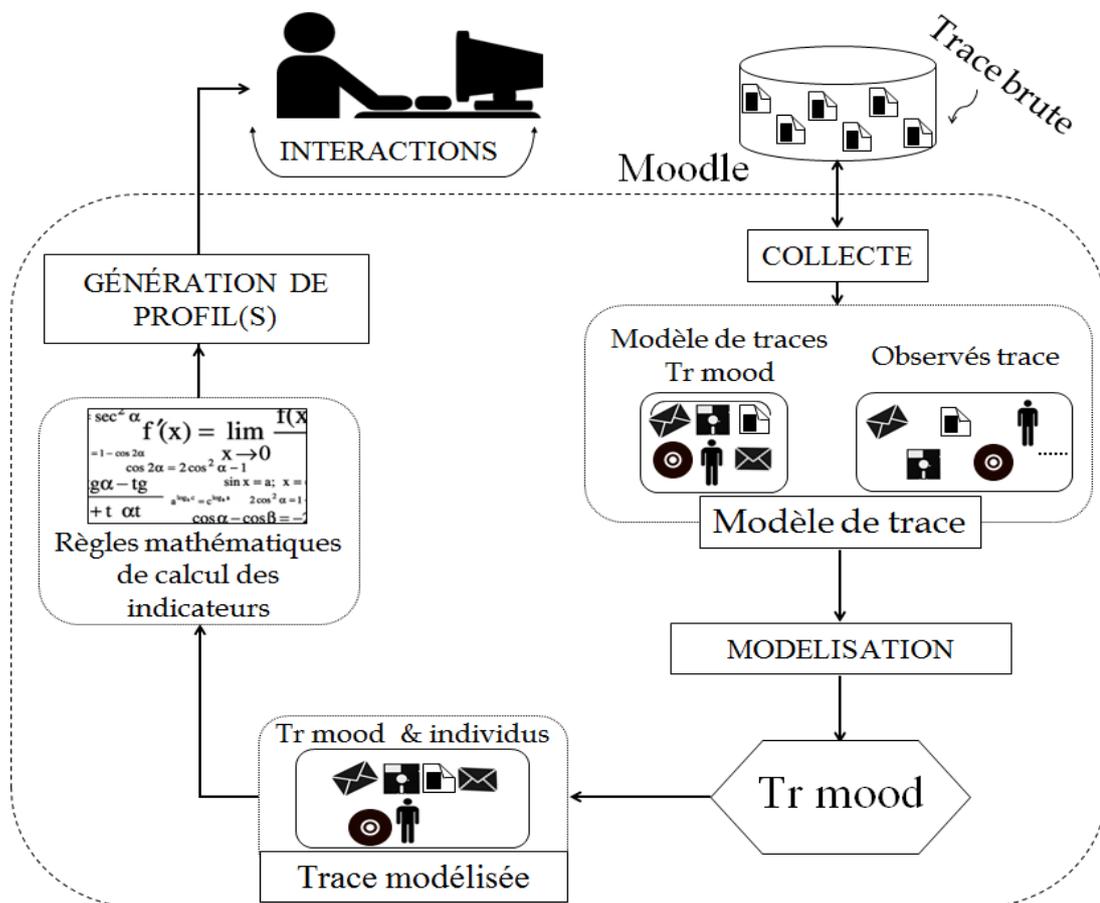


Figure III.1 : Architecture du système

La figure montre l'architecture du système dans son ensemble en explicitant le processus de manière général quant à la génération des profils, elle s'entame par une collecte de données depuis la base de données « Moodle », puis matérialise les traces modélisées selon un modèle de traces avant de passer au calculs d'indicateurs qui nous permettront au final de pouvoir générer un ensemble de profils.

III. Modélisation du système

Pour la modélisation de « GenProApp » nous avons utilisé la méthode U.P. (Unified Process), on entame la méthode par la confection des diagrammes suivants : « cas d'utilisation », de « séquence » ainsi que celui des « classe ».

➤ Diagramme de cas d'utilisation

Comme il est illustré ci-dessous, le système comporte deux acteurs : un « principal » qu'est *l'enseignant* et l'autre acteur qu'est *l'administrateur*.

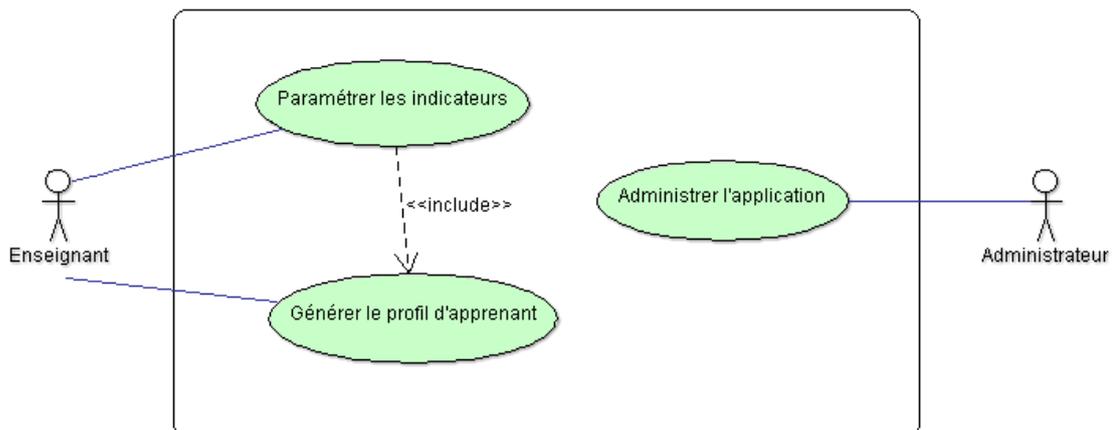


Figure III.2 : Diagramme de « cas d'utilisation »

- Administrateur : Ses interactions avec le système consistent uniquement à : «effectuer des suivis pour l'application».
- L'enseignant : Quant à l'acteur principal, il interagit en « paramétrant les indicateurs » qui contribueront à sa seconde fonction (la plus importante) : « générer les profils des apprenants ».

➤ Diagramme de séquence

Ce type de diagramme démontre son utilité par rapport au fait qu'il décrit de manière singulière et méticuleuse : le déroulement de chaque interaction d'un acteur du système par le biais des différents scénarios propre à l'action de l'acteur en question.

Le diagramme de séquence illustré dans la figure suivante prend en charge 2 cas d'utilisation : la modélisation des traces ainsi que la génération de profil de l'apprenant.

Ce diagramme comprend l'acteur qu'est « l'enseignant » ainsi que les classes suivantes : « Traces brutes », « Traces modélisées », « TrMood », « Paramètre », « Indicateur », et « Profil ».

On décrit la séquence comme il suit :

L'enseignant sollicite le système via l'interface qui lui permettra de choisir un cours, une procédure de collecte de traces à partir de la BDD « Moodle » se déclenche et une phase de modélisation de traces s'enclenchera ensuite, ce qui retournera un affichage sur l'interface de l'application offrant la possibilité à l'enseignant de faire un choix d'un cours et pouvoir le paramétrer (mode avancé).

Ensuite, le calcul des indicateurs se fera automatiquement puis, retournera sur l'interface de l'application les profils d'apprenants adhérents au cours.

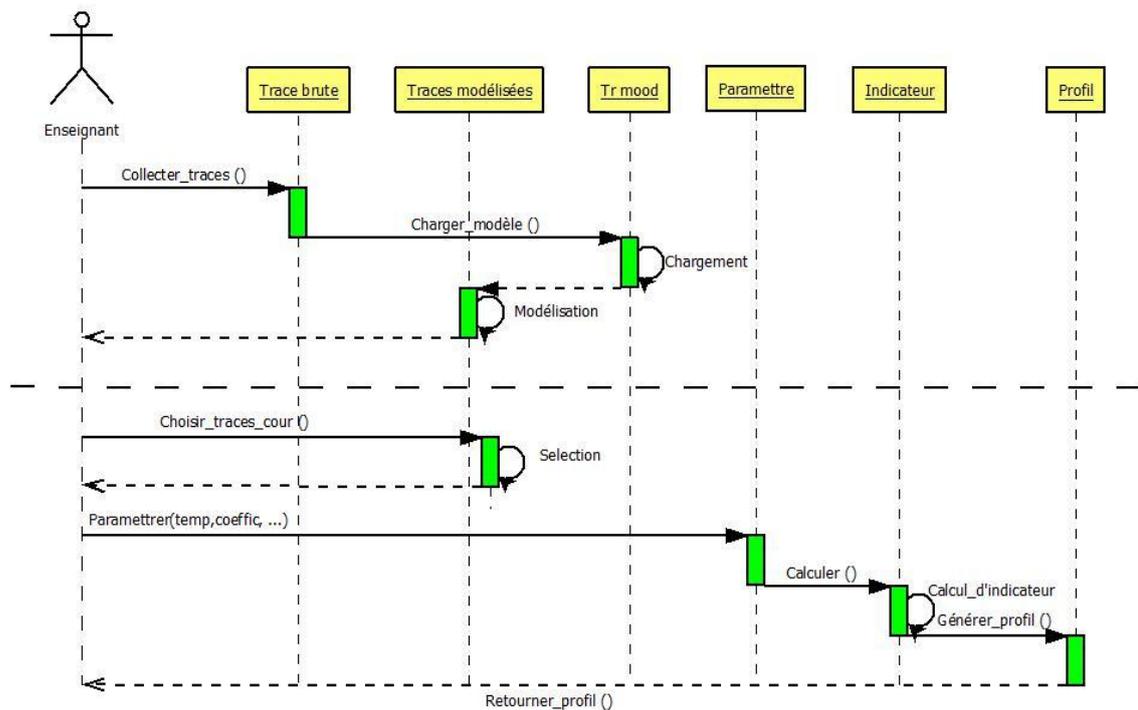


Figure III.3 : Diagramme de « séquence »

➤ Diagramme de classe

Un diagramme de classe représente les associations propres aux concepts identifiés et jugés importants dans une application. Ces concepts prennent la forme d'un ensemble de classes (dans cette application), la figure III.4 le représente.

Le diagramme de classes illustré dans la figure III.4 contient 5 classes : indicateurs, traces, « TrMood », paramètre et profil. La classe trace modélise les traces selon le model « TrMood », la classe indicateur récupère ces traces et utilise les 2 classes paramètre et profil pour calculer les indicateurs et pouvoir générer le profil des apprenant.

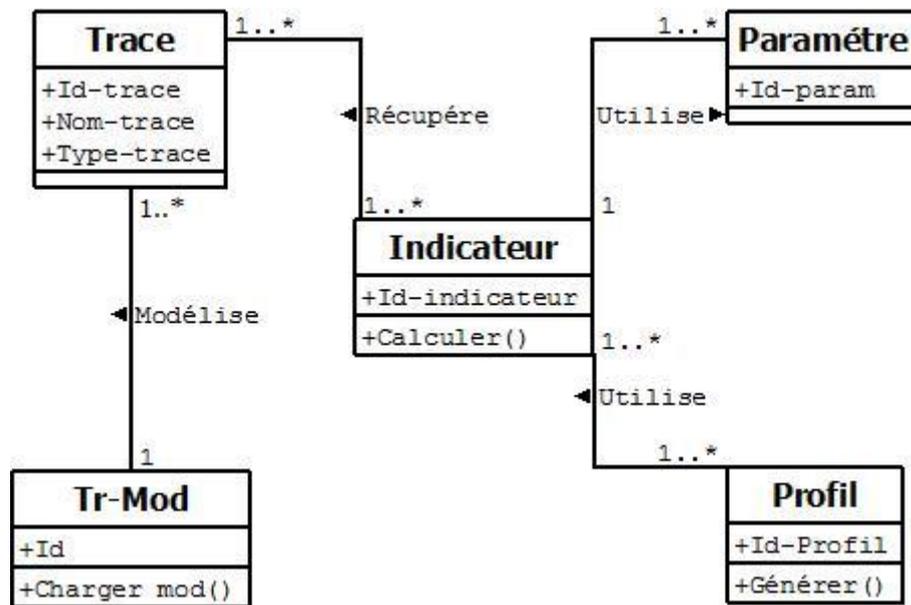


Figure III.4 : Diagramme de « classes »

IV. Processus de génération à partir des traces modélisées

Dans cette application, il existera deux options pour pouvoir générer un profil, chacune différera dans son traitement :

- La première étant « par défaut » c'est-à-dire qu'elle s'opérera en calculant deux indicateurs (définis dans le chapitre précédent) ; le premier pour « quizz » et le second pour « lesson », une fois ces calculs effectués, on prélève les notes propres aux modules pour lesquels ces indicateurs ont été calculés puis on établit une moyenne via un calcul arithmétique.
- La seconde étant « avancée », elle prend en charge deux indicateurs supplémentaires que le mode « par défaut » que sont : « course » et « forum », avec en sus une prise de considération des paramètres internes aux modules et qui influenceront le calcul ; comme le « temps optimal » et « le coefficient » (ces paramètres sont absents du mode « par défaut ») et qui seront librement saisis par l'enseignant.

Ce processus s'entame dès la modélisation des traces

- Il consiste à donner aux indicateurs et de manière simultanée :
 - Les paramètres saisis par l'enseignant.
 - Les données à partir de l'ontologie OWL via des requêtes SPARQL.

Chapitre III : Génération du profil d'apprenant

- Le calcul des quatre indicateurs retournera des valeurs (calculs détaillés dans le chapitre précédent).
- Ces notes seront pondérées par multiplication par le biais des coefficients saisies par l'enseignant.
- La note globale de l'apprenant traité sera générée à partir des opérations de l'étape précédente.
- Une fois la note calculée, un processus de notation (norme) sera sélectionné par l'enseignant.

En fonction de la note obtenue et de sa translation via la norme choisie ; un profil sera généré pour l'apprenant ayant cette note.

➤ Requêtes SPARQL

L'ensemble des indicateurs formalisés dans le chapitre précédent ont recours à des requêtes SPARQL, on donne un exemple d'une requête SPARQL utilisé par l'application « GenProApp » :

Dans la requête SPARQL suivante, il est question de sélectionner la note et la durée passée dans le module « quizz » d'un cours et d'un apprenant spécifiés.

Il faut au préalable déclarer tous les espaces de nom utilisés. Pour déclarer ces derniers on utilise le mot clé PREFIX.

```
PREFIX ns:<http://www.semanticweb.org/zenati/ontologies/2013/3/untitled-ontology-8#>
```

```
SELECT ?time ?note
```

```
WHERE { ?QA ns:hasUser ?user.
```

```
?QA ns:hasCourse ?cours.
```

```
?QA ns:hasTool ?quizz.
```

```
?QA rdf:type ns:lessonAnswer.
```

```
?QA ns:hasGrade ?grade.
```

```
?QA ns:hasTime ?times.
```

```
?times ns:inTime ?time.
```

```
?grade ns:hasNote ?note.
```

```
?user ns:hasName ?name.
```

```
filter (?cours =ns:cour && ?user = ns:user)}
```

```
GROUP BY ?time ?note
```

V. Cas d'application

Afin de tester l'application, un corpus de traces (réelles) à partir d'activités d'apprentissage est constitué, cette application réalisera entre autre les transformations de traces, les calculs et tests des indicateurs.

Une situation d'apprentissage basée sur la plateforme « Moodle » a été observée et tracée tout au long de son déroulement avec un groupe d'étudiants en Master 2.

Nous avons proposé dans chaque séance un scénario pédagogique, où les étudiants se devaient de :

- Lire l'énoncé du cours.
- Tester leurs aptitudes en compréhension grâce au module « lesson ».
- Solliciter le module respectif aux « forums » en participant aux activités proposées par ce module.
- Répondre aux interrogations du module « quizz ».

L'expérimentation faite, il est temps pour l'enseignant de tester l'application.

L'interface principale contient ce qui suit :

- Voir le cours : ce bouton charge les cours disponibles de la base de données sur la plateforme.
- Actualiser : ce bouton met à jour les traces ajoutées par des apprenants.

L'enseignant va sélectionner à titre d'exemple « voir le cours » et choisir « XML » comme dans la figure suivante :

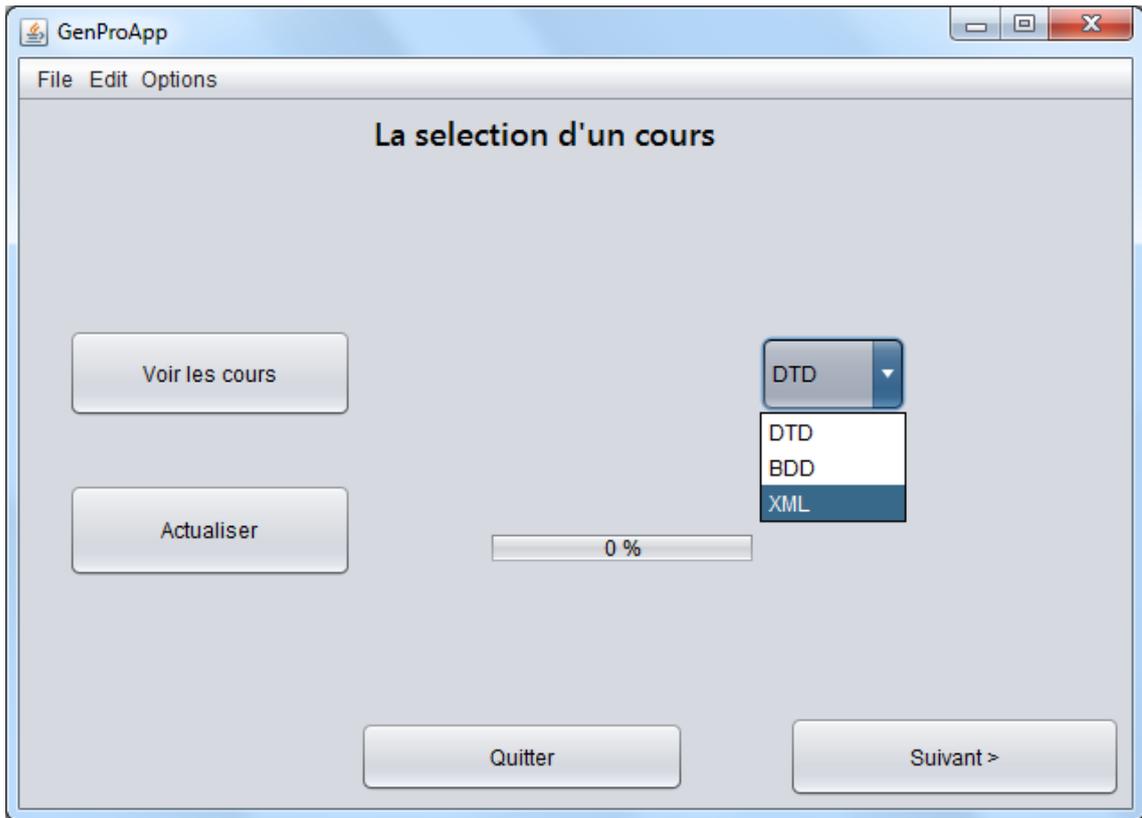


Figure III.5 : L'interface principale de « GenProApp »

L'enseignant validera son choix par le bouton « suivant » avant de devoir choisir entre les modes offerts par l'application pour générer les profils, à savoir les modes « par défaut » ou « avancé ».

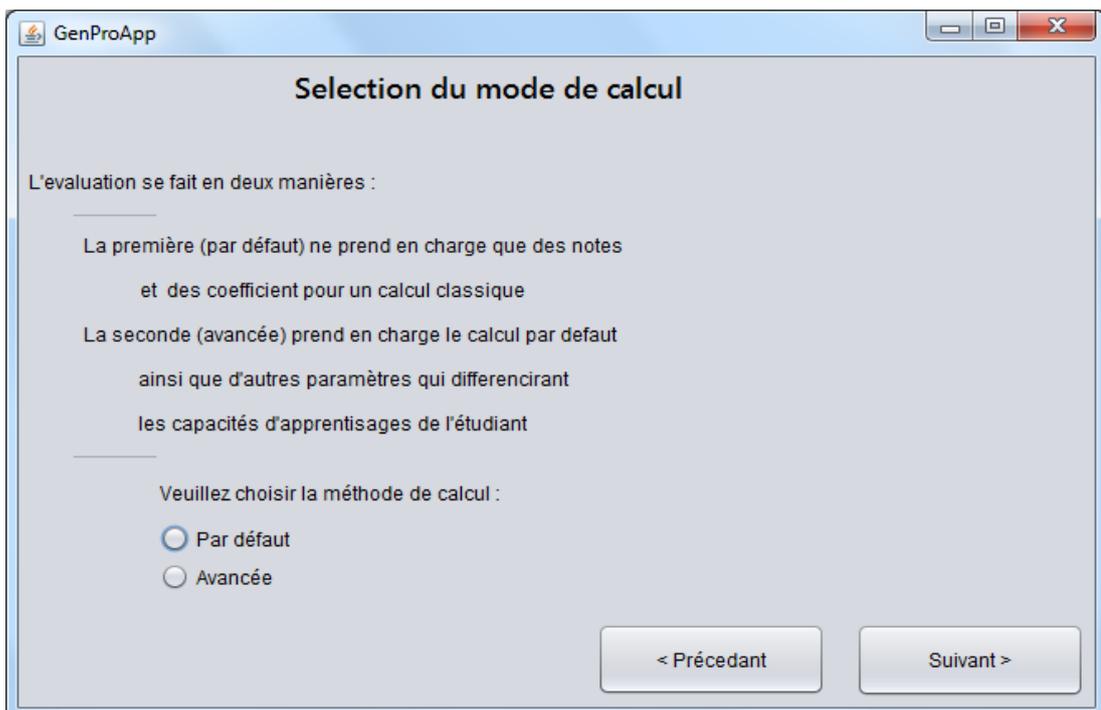
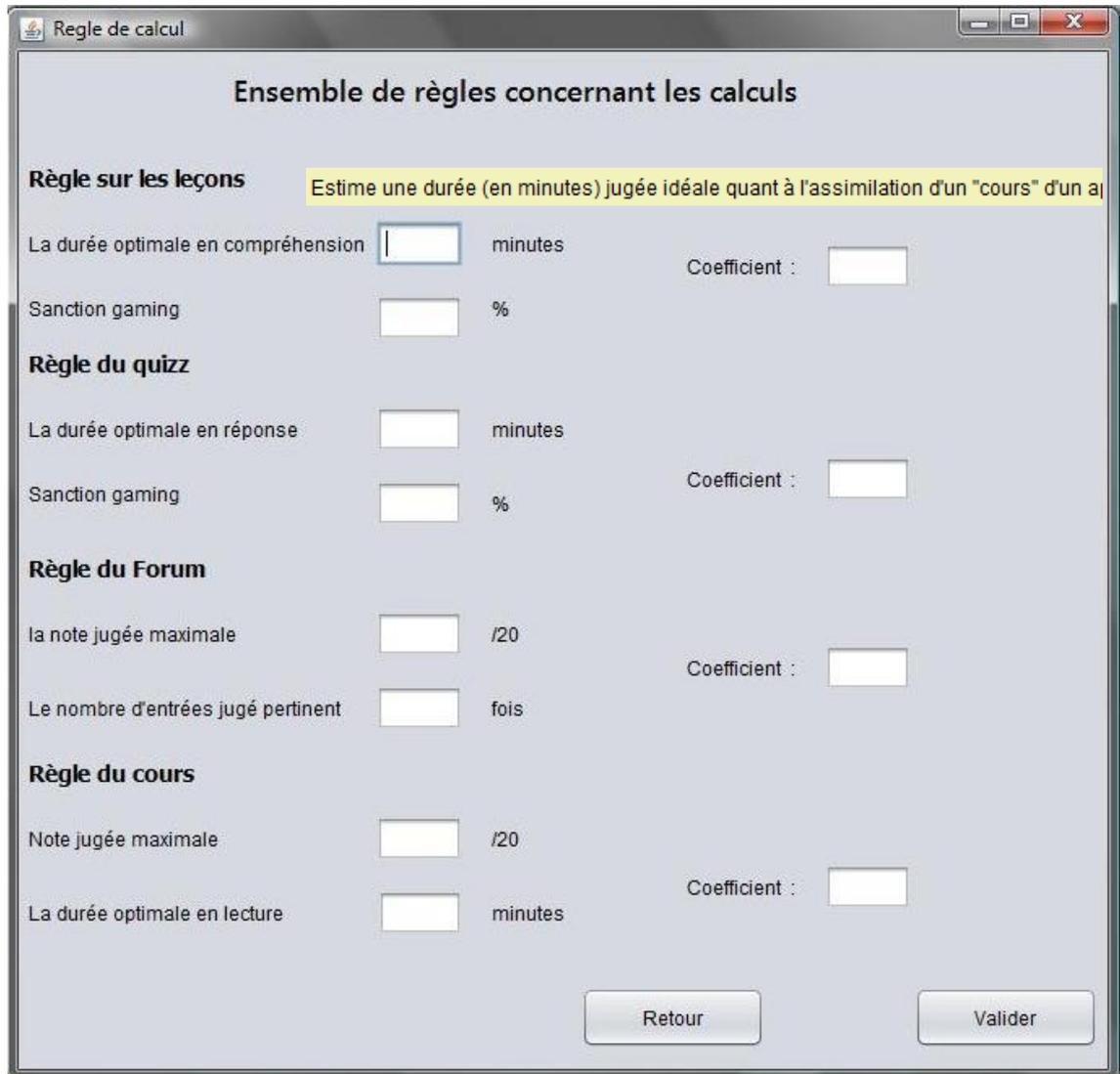


Figure III.6 : L'interface de la sélection du mode de calcul

Chapitre III : Génération du profil d'apprenant

Supposons que le mode « avancé » soit choisi:

- L'enseignant se doit de saisir des paramètres qui influenceront le déroulement du processus quant à générer le profil d'un apprenant.



The screenshot shows a window titled "Regle de calcul" with the subtitle "Ensemble de règles concernant les calculs". It contains four sections, each with a title and a description:

- Règle sur les leçons**: Estime une durée (en minutes) jugée idéale quant à l'assimilation d'un "cours" d'un a...
 - La durée optimale en compréhension: minutes
 - Sanction gaming: %
 - Coefficient:
- Règle du quizz**:
 - La durée optimale en réponse: minutes
 - Sanction gaming: %
 - Coefficient:
- Règle du Forum**:
 - la note jugée maximale: /20
 - Le nombre d'entrées jugé pertinent: fois
 - Coefficient:
- Règle du cours**:
 - Note jugée maximale: /20
 - La durée optimale en lecture: minutes
 - Coefficient:

At the bottom, there are two buttons: "Retour" and "Valider".

Figure III.7 : L'interface des règles de calculs

Le menu suivant permettra de choisir l'échelle d'évaluation telle qu'il est montré dans la figure suivante. Il est à noter que le mode « par défaut » pointe directement vers le choix de l'échelle de l'évaluation vu qu'il ne requiert pas de paramétrages.

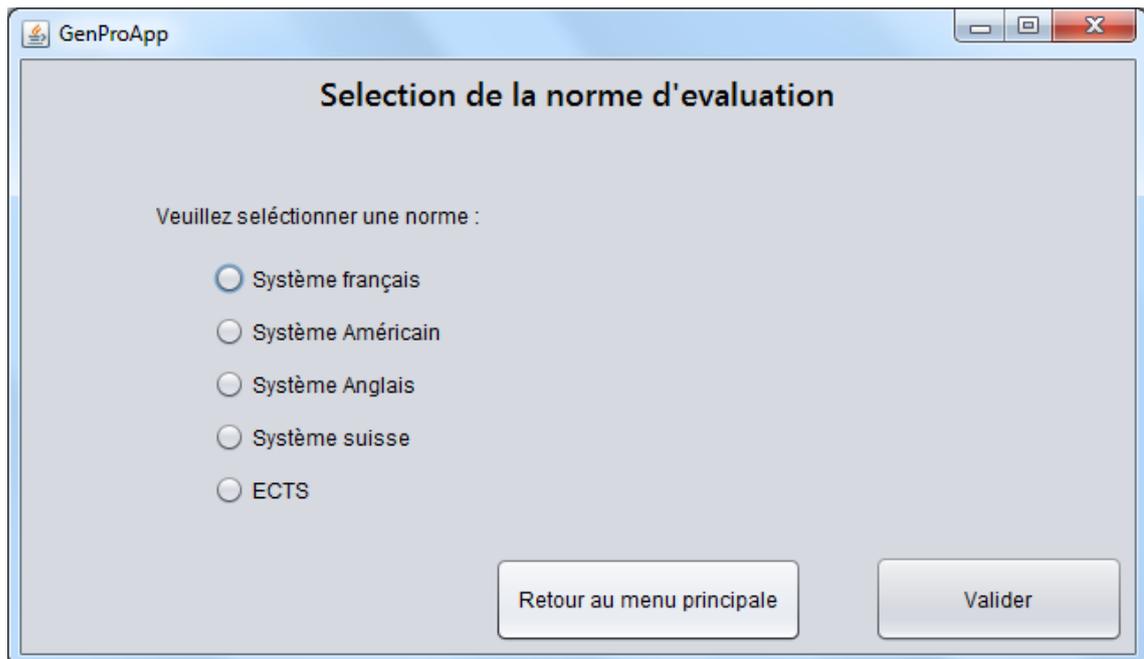


Figure III.8 : L'interface qui permet la sélection de l'échelle de notation

Après avoir préparé l'application au traitement, il est intéressant d'observer les résultats qui ont été retournés dans les deux modes stipulés.

La figure suivante est une capture d'image des résultats du mode « par défaut ».

La génération du profil

Cours	user Name	note	la mention	definition
XML	salah13	15.0	mention bien	Résultats supérieurs à la moyenne, malgré un certain nombre d'insuffisances
XML	imanova	18.0	mention très bien	Résultats remarquables, avec seulement quelques insuffisances mineures
XML	sofyane46	17.0	mention très bien	Résultats remarquables, avec seulement quelques insuffisances mineures
XML	adela35	15.0	mention bien	Résultats supérieurs à la moyenne, malgré un certain nombre d'insuffisances
XML	meriemova	9.0	oral au rattrapage	Les résultats satisfont aux critères minimaux
XML	hakimous	16.0	mention très bien	Résultats remarquables, avec seulement quelques insuffisances mineures
XML	yasser13	15.0	mention bien	Résultats supérieurs à la moyenne, malgré un certain nombre d'insuffisances
XML	ouss23	7.0	échec définitif	Un travail supplémentaire est nécessaire pour l'octroi d'un crédit
XML	hadilova	5.0	échec définitif	Un travail supplémentaire est nécessaire pour l'octroi d'un crédit

< Retour Fermer

Figure III.9 : Génération des profils d'apprenants dans le mode « par défaut »

La figure montre différents mentions et résultats démontrant une certaine pertinence quant au succès de l'application pour générer le profil dans le mode « par défaut ».

Chapitre III : Génération du profil d'apprenant

Pour pouvoir comparer les modes « par défaut » et « avancé » sur une même échelle, on se doit d'amputer le mode « avancé » des deux indicateurs supplémentaires que sont « cours » et « forum » en annulant leurs coefficients lors des paramétrages précédant les calculs en question.

Utilisateur	Imanova	Sofyane46	Hakimous	Merimeova	Ouss23	Salahe13
NoteQuizz	18	17	16	9	7	15
TimeQuizz	10	25	30	30	40	60
NoteLeçon	18	17	16	9	7	15
TimeLeçon	20	50	60	60	80	120
TimeQuizzOpt	30	30	30	30	30	30
TimeLeçonOpt	60	60	60	60	60	60

Tableau III.1 : Table recensant les notes des différents utilisateurs avec un paramétrage de temps

Utilisateur	noteGenParDefault	noteGenAnvacé	Situation
Imanova	18	10,12	Gaming en leçon et quizz
Sofyane46	17	17	Doué
Hadilova	5	2,81	Gaming en leçon et quizz
Hakimous	16	16	temps optimale leçon et quizz
Merimeova	9	9	temps optimale leçon et quizz
Ouss23	7	6,86	il a dépassé le temps optimale par 10 minutes
Adela35	15	14,9	il a dépassé le temps optimale par 5 minutes
Yasser13	15	14,22	il a dépassé 20 minutes
Salahe13	15	13,75	il a dépassé 30 minutes

Tableau III.2 : Comparaison des notes générées par les modes de « GenProApp » à partir du tableau III.I

Les notes générées en mode « avancé » diffèrent de manière apparente par rapport à ceux du mode « par défaut », certains cas le démontrent clairement :

- L'utilisateur « Imanova » a obtenu la meilleure note selon une notation classique dans le mode « par défaut », après avoir impliqué les paramètres temporels par l'enseignant, la note à « Imanova » s'est vu chuter de presque 8 points, car elle est en situation de gaming, il est à noter que la sanction écopée lors d'une situation de gaming était fixée à 56% .
- L'utilisateur « Hakimous » a vu sa note inchangée car son temps concret lors d'un quiz et d'une leçon s'est révélé quasi identique aux vœux de l'enseignant.
- L'utilisateur « Salahe13 » écope d'une sanction de retard, ce qui a diminué sa note.

Chapitre III : Génération du profil d'apprenant

- Il est à noter que la note obtenue après une sanction de retard est proportionnellement relative au temps.

Voici un autre exemple de pratique, celui-ci recense tous les calculs possibles du mode « avancé », à savoir les quatre indicateurs.

La figure suivante prend en charge les mêmes utilisateurs, les coefficients des modules « quiz », « lesson », « cours » et « forum » sont respectivement : « 4 », « 3 », « 2 » et « 1 ».

Les notes maximales considérées aux modules « forum » et « lesson » sont identiques : 20.

Utilisateur	Imanova	Sofyane46	Hadilova	Hakimous	Merimeova	Ouss23
NoteQuizz	18	17	5	16	9	7
TimeQuizz	10	25	10	30	30	40
NoteLeçon	18	17	5	16	9	7
TimeLeçon	20	50	20	60	60	80
forumView	3	2	1	1	4	5
courseView	36	4	4	8	3	19
TimeQuizzOpt	20	20	20	20	20	20
TimeLeçonOpt	40	40	40	40	40	40
forumViewOpt	3	3	3	3	3	3
courseViewOpt	10	10	10	10	10	10

Tableau III.3 : Table recensant les notes des différents utilisateurs avec un paramétrage de temps

Utilisateur	Imanova	Sofyane46	Hadilova	Hakimous	Merimeova	Ouss23
noteGenParDefault	18	17	5	16	9	7
noteGenAnvacé	12,31	15,83	5,34	15,85	10,28	10,09

Tableau III.4 : Comparaison des notes générées par les modes de « GenProApp » à partir du tableau III.3

Les résultats qui figurent dans la table nous intéressent par rapport au cas d'un utilisateur précis:

La note de l'utilisateur « Ouss23 » a augmenté en mode « avancé » par rapport à ce qu'il a obtenu par rapport au mode « par défaut », cela s'explique par le fait qu'il ait participé aux deux modules qui ne sont pas pris en considération dans le mode « par défaut » que sont « cours » et « forum ».

VI. Conclusion

Ce chapitre a été consacré à la modélisation et réalisation du système « GenProApp ». Les différentes étapes de réalisation ont été détaillées et un exemple d'utilisation d'indicateur a été donné via l'utilisation d'une requête SparQL. Nous avons terminé le chapitre par dérouler un scénario d'utilisation de l'application avec l'explication des résultats obtenus. Nous avons pu observer les résultats qui ont été retournés dans les deux modes stipulés. Les profils générés à partir de l'application étaient correct dans les sens où ils étaient cohérents avec les règles de calculs établies et les paramètres choisis. De ce fait le système répond aux besoins définis et obéit aux règles et indicateurs utilisés.

Conclusion générale

Le travail présenté ayant pour objectif la génération du profil d'apprenant(s) en session d'apprentissage sur la plateforme Moodle en se basant sur les traces s'est révélé être satisfaisant grâce à l'application mise en œuvre.

Pour y avoir parvenu, nous avons mobilisés le potentiel des ontologies et leurs affectations sur les options proposées sur la plateforme Moodle, leurs intérêts se sont révélés certains quant à la sémantique des données.

L'application « GenProApp » se veut ergonomique ; elle facilite la tâche de l'enseignant en lui donnant une possibilité pour pouvoir générer des profils d'apprenants d'une manière automatisée en fonction des seuils et des coefficients paramétrables, et cela selon des normes d'évaluations officielles.

Perspectives

Nous souhaiterions améliorer l'expérience de l'application « GenProApp » par ce qui suit :

- Améliorations de la fonction de génération de profil d'apprenant sur la plateforme Moodle via un plugin qui rendra le traitement centralisé sur la plateforme.
- L'ajout de nouveaux indicateurs selon les besoins d'observations exprimés par l'enseignant.

Liste de figures

Figure I.1 : CourseVis [Mazza et Dimitrova 04]	9
Figure I.2 : Architecture de CourseVis [Mazza et Dimitrova 04]	10
Figure I.3 : ClassroomVis : Une visualisation des états des apprenants, les parcours effectués et les activités réalisées [France et al. 07]	11
Figure I.4 : Extraction des données Moodle [Romero, Ventura & Garcia 08].....	12
Figure I.5 : Architecture du système.....	13
Figure II.1 : Le méta-modèle de la trace première en format UML	22
Figure II.2 : Instancier le méta-modèle pour obtenir le modèle de la trace première Moodle (en format UML)	23
Figure II.3 : Copie écran Protégé : concepts du modèle d'utilisation correspondant à la trace première.....	24
Figure II.4 : Copie écran Protégé : propriété d'objet du modèle d'utilisation	25
Figure II.5 : Copie écran Protégé : propriété de données du modèle d'utilisation.....	25
Figure II.6 : Copie écran Protégé : L'ontologie OWL visualisée avec OntoGraf	26
Figure II.7 : Un exemple d'observé écrit en OWL, généré par notre Application	27
Figure III.1 : Architecture du système	33
Figure III.2 : Diagramme de « cas d'utilisation ».....	34
Figure III.3 : Diagramme de « séquence ».....	35
Figure III.4 : Diagramme de « classes »	36
Figure III.5 : L'interface principale de « GenProApp »	39
Figure III.6 : L'interface de la sélection du mode de calcul	39
Figure III.7 : L'interface des règles de calculs	40
Figure III.8 : L'interface qui permet la sélection de l'échelle de notation	41
Figure III.9 : Génération des profils d'apprenants dans le mode « par défaut »	41

Liste des tableaux

Tableau I.1 : Synthèse sur les différents travaux de visualisation de la trace, et identification de nos besoins par rapport à ces travaux.....	14
Tableau II.1 : La table mdl_user	17
Tableau II.2 : La table mdl_course	18
Tableau II.3 : La table mdl_log	18
Tableau II.4 : La table mdl_lesson.....	18
Tableau II.5 : La table mdl_lesson_pages	18
Tableau II.6 : La table mdl_lesson_timer	19
Tableau II.7 : La table mdl_lesson_grades	19
Tableau II.8 : La table mdl_quizz	19
Tableau II.9 : La table mdl_quizz_attempts	19
Tableau II.10 : La table mdl_quizz_grades	20
Tableau II.11 : La table mdl_chat	20
Tableau II.12 : La table mdl_chat_messages.....	20
Tableau II.13 : La table mdl_forum.....	20
Tableau II.14 : La table mdl_forum_discussions.....	21
Tableau II.15 : La table mdl_forum_posts.....	21
Tableau II.16 : Les systèmes de notation.....	31
Tableau III.1 : Table recensant les notes des différents utilisateurs avec un paramétrage de temps.....	42
Tableau III.2 : Comparaison des notes générées par les modes de « GenProApp » à partir du tableau III.I	42
Tableau III.3 : Table recensant les notes des différents utilisateurs avec un paramétrage de temps.....	43
Tableau III.4 : Comparaison des notes générées par les modes de « GenProApp » à partir du tableau III.3.....	43

Références Bibliographiques

- [1] : Tchounikine P. (2002), Pour une ingénierie des Environnements Informatiques pour l'Apprentissage Humain. In: Revue I3 Information Interaction Intelligence Vol. 2, n°1, p. 59-95.
- [2] : O'Neil M. What is E-learning, Home Magazine, May 2001.
- [3] : Innovative Learning Technologies. Définition des termes LCMS et LMS : LCMS et LMS, quelles différences ?,
<http://www.innovativelearningtechnologies.fr/tendances/definition-des- termes-lcms-et-lms-lcms-et-lms-quelles-differences-202>, consulté le 20 Mai 2013.
- [4] : SETTOUTI Lotfi-Sofiane, PRIE Y, MARTY Jean-Charles, MILLE Alain. Vers des systèmes à base de traces modélisées pour les E.I.A.H.. Workshop Traces, interactions, co-constructions collectives et relations à la cognition, AS CoMETE, Paris, Avril 2007.
- [5] : Tarek Djouad, Lotfi Sofiane Settouti, Alain Mille, Yannick Prié et Christophe Reffay (2010). Un Système à Base de Traces pour la modélisation et l'élaboration d'indicateurs d'activités éducatives individuelles et collectives. Mise à l'épreuve sur Moodle. Techniques et Science Informatique, 29 (6), Hermès, pp 721-741.
- [6] : DIMITRACOPOULOU, A., BRUILLARD, E. (2006). Enrichir les interfaces de forums par la visualisation d'analyses automatiques des interactions et du contenu. Revue STICEF, Vol. 13.
- [7] : Cram D., « Visualisation de Traces : Application aux Traces Réflexives d'eLycée », Rapport de Master de recherche, Université Claude Bernard Lyon1, 2007.
- [8] : Mazza R., Dimitrova V., « Visualising Student Tracking Data to Support Instructors in Web-Based Distance Education », Dans les actes de: the thirteenth International World Wide Web Conference-Educational Track, New York, USA, 2004, p.154-161.
- [9] : France L., Heraud J M., Marty J C., Carron T., « Visualisation et régulation de l'activité des apprenants dans un EIAH tracé », dans les Actes de la conférence EIAH 2007, Lausanne, Suisse, 27-29 Juin 2007, p.197-184.

- [10] : C. Romero, S. Ventura, E. Garcia (2008). "Data Mining in Course Management Systems: MOODLE Case Study and Tutorial". Computers & Education. 51(1): 368–384.
- [11] : Djouad T., Ingénierie des indicateurs d'activités à partir de traces modélisées pour un Environnement Informatique d'Apprentissage Humain (THESE). Université Claude Bernard - Lyon I.
- [12] : DJOUAD, T. (2008). Analyser l'activité d'apprentissage collaboratif : Une approche par transformations spécialisées de traces d'interactions. 2ième rencontre des jeunes chercheurs RJC-E.I.A.H.08, Lille, France, p. 93-98.
- [13] : Université de Nîmes. ECTS grading scale. En ligne http://www.unimes.fr/en/practical_information/ects_grading_scale.html, consulté le 5 juin 2013.