



DNE - GTnum2 Learning Analytics - Etat de l'art sur les outils et méthodes issus de la recherche française

Marie Lefevre, Sébastien Iksal, Julien Broisin, Olivier Champalle, Valérie Fontanieu, Christine Michel, Amel Yessad

► To cite this version:

Marie Lefevre, Sébastien Iksal, Julien Broisin, Olivier Champalle, Valérie Fontanieu, et al.. DNE - GTnum2 Learning Analytics - Etat de l'art sur les outils et méthodes issus de la recherche française. [Rapport de recherche] Ministère de l'éducation nationale. 2018. hal-02453676

HAL Id: hal-02453676

<https://hal.science/hal-02453676>

Submitted on 24 Jan 2020

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

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

Thème 2 : Learning Analytics

Etat de l'art sur les outils et méthodes issus de la recherche française

Auteurs : Marie Lefevre et Sébastien Iksal (coordinateurs), Julien Broisin, Olivier Champalle, Valérie Fontanieu, Christine Michel, Amel Yessad.

Date : Décembre 2018

Introduction	2
1 - Présentation de l'étude	2
2 - Présentations du résultat de l'étude selon les étapes de l'analyse	4
Construire le jeu de traces	4
Etape de collecte	5
Modèles de traces	5
Analyser les jeux de traces	12
Outils généralistes de statistique	12
Algorithmes automatiques	14
Langage dédié : du langage informatique à la langue naturelle	15
Outils graphiques	16
Visualiser les résultats de l'analyse	18
Outils d'exploration généralistes	19
Outils type "tableau de bord" dédiés aux EIAHs	21
3 - Synthèse des outils étudiés	25
4 - Présentation des principaux outils	26
DDART	26
Laalys	29
UTL - Usage Tracking Language	33
Lab4CE	36
KTBS4LA	40
D3KODE	41
EMODA	45
Perspectives	47
Références	48
Annexe 1 : Liste des outils étudiés	52

Introduction

Les Learning Analytics, ou l'analyse des traces d'apprentissage, sont définis comme « l'évaluation, l'analyse, la collecte et la communication des données relatives aux apprenants, leur contexte d'apprentissage, dans la perspective d'une compréhension et d'une optimisation de l'apprentissage et de son environnement » [Long et al., 2011].

L'analyse des traces d'apprentissage suit un cycle passant par les étapes de **collecte des traces, d'analyse de ces traces et d'exploitation (souvent de la visualisation)** des traces et des indicateurs produits par l'analyse [Fayyad et al., 1996] [Clow, 2012] [Stamper et al., 2011].

La **collecte** des données est une étape primordiale car il s'agit de récupérer toutes les données numériques représentant l'activité des usagers afin de mener un processus d'analyse et d'obtenir un reflet du déroulement des situations d'apprentissage. Ainsi la collecte des données concerne l'observation de l'apprenant et le **traçage** de ses interactions médiées par les outils, le **stockage** des traces récoltées et l'**import** de traces dans les outils d'analyse. Ces différentes actions peuvent être faites par un unique outil de collecte et d'analyse ou peuvent être réparties dans différents outils à combiner pour mener à bien les analyses.

L'**analyse** des traces consiste à manipuler les données pour essayer d'extraire des informations. Certaines analyses vont nécessiter des **pré-traitements** sur les données pour les mettre en forme, les nettoyer, vérifier leur fiabilité, etc.

Ces étapes sont parfois complétées par une étape de **partage** des traces, des processus d'analyse, des indicateurs et/ou des visualisations produites.

Tout au long de ce cycle plusieurs acteurs entrent en jeu [Greller et al., 2012] : **les apprenants, les équipes techniques et pédagogiques, les institutions, les familles et enfin les chercheurs**. Ces acteurs ont différents rôles lors de l'analyse et selon les contextes et les outils d'analyse. Les apprenants sont tantôt sujet de l'observation pour produire des traces, tantôt ils prennent le rôle d'analyste en utilisant des outils réflexifs permettant de comprendre leurs traces, de même pour les enseignants.

Le document présente une étude des outils et méthodes proposés dans le cadre des Learning Analytics par la communauté de recherche Française.

1 - Présentation de l'étude

Notre étude a porté sur 34 outils ou méthodes issus ou utilisés dans le cadre des projets de Recherche en France que l'ensemble des participants a pu identifier, il est possible que certains outils soient absents, probablement en raison de l'absence ou quasi absence de publications. La première partie de l'étude a consisté à établir une liste de ces outils en précisant différentes caractéristiques connues pour chacun. L'une des finalités porte sur l'identification de catégories et/ou de regroupements selon des caractéristiques communes.

L'autre se focalise sur la mise en correspondance de ces outils avec les différentes étapes classiques d'un cycle d'analyse de données : la collecte, le stockage, l'analyse et la visualisation. En parallèle, nous souhaitons mettre en évidence les aspects liés au partage et à l'interopérabilité de ces outils.

Les différentes caractéristiques étudiées sont :

- Le nom de l'outil
- L'éditeur (Université, entreprise, projet ...)
- Les informations de contact (personne et adresse mail)
- La licence
- L'accessibilité (au sens, l'outil est-il téléchargeable facilement)
- L'existence d'une documentation, si oui, le lien de téléchargement
- Le type d'outil (API, Plateforme, Logiciel, Plugin, Méthode, ...)
- L'objectif principal de l'outil (Collecte, Analyse, Partage, Visualisation, Méthode...)
- Une description
- La liste des fonctionnalités
- Les particularités (en quoi l'outil est-il innovant)
- Les limites
- Le public cible
- Le degré de maturité selon l'échelle des TRL¹
- L'état de l'outil (projet de recherche, prototype, outil industriel)
- Le ou les systèmes d'exploitation compatibles
- L'interopérabilité
- Les formats de données en entrée
- Les formats de sortie
- Les articles scientifiques associés (liens zotero)
- Quelques illustrations
- Les cas d'usage connus avec leurs descriptions

Nous avons ensuite écarté les outils qui n'étaient plus disponibles ou ceux qui étaient devenus trop obsolètes pour être exploités. Nous sommes arrivés à une liste réduite de 18 outils et méthodes que nous avons pu classer selon les différentes étapes du cycle d'observation et selon le public visé par ces outils. Nous avons identifié les rôles des personnes amenées à prendre en main l'outil (un analyste, un développeur, un décideur), ainsi que les rôles des personnes destinataires des résultats d'analyse (Apprenant, Enseignant, Administratif). La liste complète des outils étudiés se trouve en annexe 1. Nous nous focalisons ici sur la liste restreinte que nous avons catégorisée selon le cycle et dont les outils seront présentés dans ce document :

- Outils permettant la collecte de données
 - D3KODE, DDART, kTBS4LA, Laalys, Lab4ce, Limesurvey, SBT-IM, Tatiana, TRAVIS, UnderTracks, UTL
- Outils permettant le stockage des données
 - Abstract, D3KODE, DDART, DisKit, dmt4sp, Laalys, Lab4ce, LEA4AP, Limesurvey, SBT-IM, Tatiana, TRAVIS, T-store, UTL

¹ <http://www.cea.fr/multimedia/Documents/infographies/Defis-du-CEA-infographie-echelle-trl.pdf>

- Outils permettant l'analyse des données
 - Abstract, D3KODE, DDART, DisKit, dmt4sp, kTBS4LA, Laalys, LEA4AP, SBT-IM, Taaabs, Tatiana, Transmute, TRAVIS, T-store, UnderTracks, UTL, EMODA
- Outils permettant la visualisation des données
 - Abstract, D3KODE, DDART, DisKit (+Transmute), dmt4sp (textuelle), kTBS4LA (+Taaabs), Laalys, Lab4ce, LEA4AP, Limesurvey, SBT-IM, Tatiana, TRAVIS, UnderTracks, UTL, EMODA
- Outils permettant le partage et l'interopérabilité
 - D3KODE, SBT-IM, Tatiana, UTL

2 - Présentations du résultat de l'étude selon les étapes de l'analyse

Présentée en introduction, l'analyse des traces d'apprentissage suit un cycle composé de plusieurs étapes : collecte, analyse et exploitation/visualisation. Dans cette section, les outils analysés sont classés et détaillés suivant ces différentes étapes.

Construire le jeu de traces

Les types de traces les plus communément considérés sont les interactions de l'utilisateur avec le dispositif de formation. Ces traces sont enregistrées automatiquement. Peu de systèmes utilisent les traces de log standard construites par les applications client-serveur car elles manquent de précision. Les systèmes implémentent plutôt des capteurs spécifiques dans les applications de formation sur la base des modèles de traces nécessaires pour réaliser les analyses. Les capteurs sont donc positionnés sur les modalités d'interaction pertinentes pour adapter le système ou construire les indicateurs qui permettent la prise de décision des utilisateurs considérés (apprenant, enseignant ou analyste).

Les informations collectées suites à ces interactions peuvent être relatives à l'interaction elle-même (quelle fonctionnalité est utilisée, à quel moment, pendant combien de temps, ...) ou aux résultats de l'interaction (contenu d'un message de chat, séquence audio ou vidéo relative à une conversation, document produit à plusieurs, ...). Les traces descriptives de l'interaction sont structurées sur la base d'un modèle alors que les traces des produits de l'activité ne le sont souvent pas.

D'autres traces, appelées déclarées ou auto-rapportées, sont aussi considérées dans les systèmes d'analyse. Elles sont renseignées manuellement par l'utilisateur. Elles correspondent à un jugement de l'utilisateur sur son activité ou sur les acteurs-partenaires de son activité. Ces traces peuvent être fermées et structurées selon un modèle, ou être ouvertes. Ces traces sont généralement textuelles.

Etape de collecte

Il existe différentes solutions pour la collecte de données selon les possibilités techniques (disponibilité ou non d'une connexion réseau) ou les besoins en terme de calcul :

- la collecte de données d'observation avec stockage dans les applications éducatives telle que Tactiléo. Ce stockage local permet de tracer sur des outils non connectés, en l'occurrence avec Tactiléo, les élèves sortent de l'établissement avec les tablettes qui tracent leurs activités, et au retour, les données peuvent être transférées vers des systèmes d'analyse évolués. cela permet aussi d'utiliser les traces avec différents systèmes d'analyse. Enfin, les traces proviennent des applications elles mêmes donc nous pouvons obtenir des informations très précises.
- la collecte de données d'observation avec stockage dans une base de traces, tel TraceMe. Cette solution délocalisée a l'avantage d'imposer un format unique, un standard que respectent tous les systèmes d'apprentissage qui envoient leurs traces. Cela permet par la suite de capitaliser et réutiliser les processus d'analyse sur différents systèmes d'apprentissage.
- la collecte de données d'observation avec stockage dans un outil d'analyse, tel Travis, T-store, DDART, UTL, D3KODE. Cette fois-ci le système d'analyse intègre directement les traces et peut donc réaliser les analyses en temps réel ce qui permet de fournir aux enseignants des informations rapidement. L'inconvénient repose sur le fait que les traces sont plus difficilement partageables entre les différents systèmes d'analyse.
- enfin, la collecte de données autres, comme les sondages de LimeSurvey. C'est une technique différente qui repose sur les réponses apportées par les différents acteurs, cela relève plus du ressenti que de l'observation réelle mais qui apporte toutefois des informations complémentaires et utiles pour améliorer la compréhension de l'observation.

Modèles de traces

Les outils présentés plus loin dans ce document s'appuient sur des modèles de données propriétaires. Parmi les nombreux modèles de traces développés par la communauté scientifique, nous décrivons dans cette section ceux ayant fait l'objet d'une adoption à grande échelle.

Attention metadata, ou **Attention.XML**, est une spécification ouverte pour tracer et partager les artefacts sur lesquels l'utilisateur a porté son attention (e.g., un document lu, regardé ou écouté par un utilisateur). La conception de Attention.XML repose sur trois prémisses : (1) les *flux* Attention correspondent à un utilisateur spécifique, (2) les enregistrements Attention représentent les *objets* sur lesquels l'utilisateur a porté son attention, et (3) les objets peuvent être collectés à partir de différentes *sources* de données.

Le schéma Attention.XML est décrit sur la partie gauche de la Figure 1. L'élément racine, *group*, comprend un titre (e.g., le nom de l'utilisateur pour lequel des objets sont collectés) et un élément *feed* qui décrit l'ensemble des sources de données (e.g., blog, site web, etc.)

fournissant des objets d'attention. Ce dernier élément est décrit par différents attributs et inclut un élément *item* représentant les objets sur lesquels l'utilisateur a porté son attention.

Toutefois, le manque de détail concernant l'usage des objets par les utilisateurs a été considéré comme un inconvénient majeur de cette initiative. Alors [Wolpers et al., 2007] ont introduit **Contextualized Attention Metadata** (CAM), une extension d'Attention.XML, afin de capturer plus finement les informations comportementales des utilisateurs. Comme l'illustre la partie droite de la Figure 1, CAM s'intéresse aux événements qui reflètent les actions se produisant sur les objets. Un événement est décrit, entre autres, par une estampille horaire et une description. Il peut être associé à une action d'un certain type et détaillée par des données associées. De plus, un événement se produit dans un certain contexte lors d'une session particulière. CAM a été adopté dans le cadre du projet Européen Open Discovery Space dont l'objectif était de concevoir un portail dédié à la recherche et à la recommandation de ressources, ainsi qu'à la constitution de communautés de pratique.

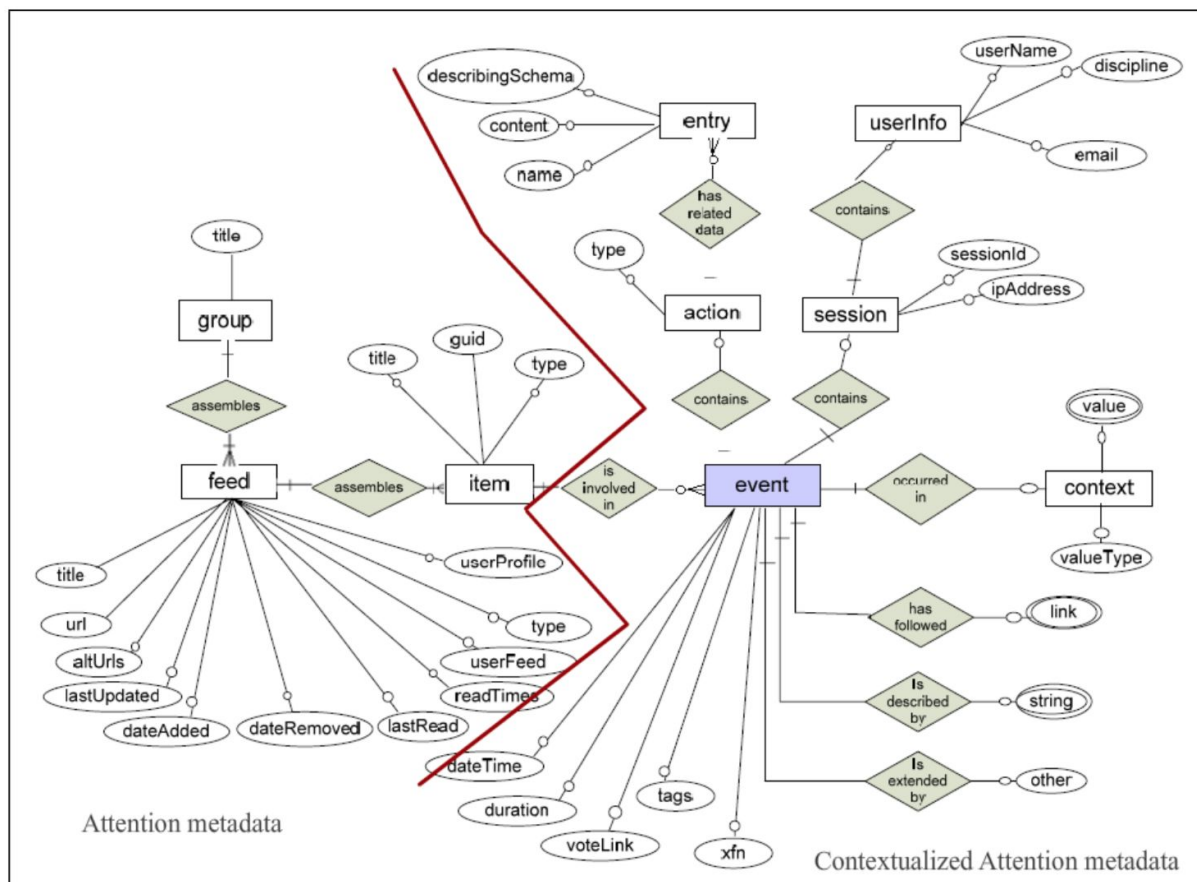


Figure 1. Le schéma CAM [Wolpers et al., 2007]

La spécification **Activity Streams** [Snell et al., 2012a] définit un format pour décrire les activités réalisées par un utilisateur sur un système ou une application ; elle n'a pas été conçue spécifiquement pour l'éducation. Un Activity Stream est une collection d'une ou plusieurs activités réalisées par un individu, et généralement exprimée au format JSON.

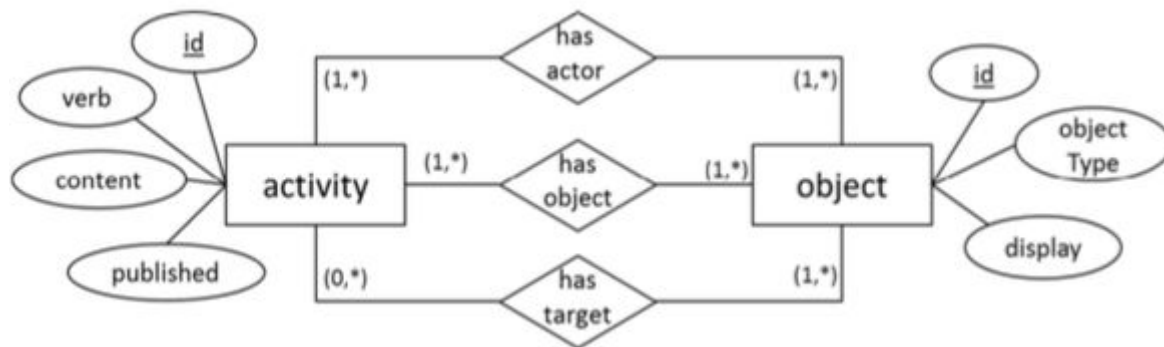


Figure 2. Représentation simplifiée du schéma Activity Streams

La Figure 2 montre les éléments principaux du schéma Activity Streams. Une activité doit au moins contenir une description de l'entité ayant réalisé l'activité (propriété *actor*) ainsi que la date et l'heure auxquelles l'activité a été publiée (propriété *published*). Le groupe de travail Activity Stream recommande qu'une activité contienne également un verbe, un objet, et un identifiant. Le verbe identifie l'action décrite par l'activité (e.g., « consulte », « évalue », « envoie »), l'objet décrit l'artefact sur lequel a été réalisée l'activité (e.g., la ressource consultée ou le message envoyé), et l'identifiant permet de retrouver une activité parmi un ensemble d'activités. La propriété *target* est optionnelle et peut être utilisée lorsqu'elle est indiquée par le verbe ; par exemple, dans l'activité « Pierre a envoyé un message à Paul », « Paul » est la cible de l'activité.

Le schéma Activity Base [Snell et al., 2012b] propose une spécification pour décrire les valeurs des propriétés *actor*, *object* et *target*, mais n'importe quel objet Activity Stream peut être étendu par d'autres propriétés qui ne sont pas définies dans cette spécification afin d'apporter autant de flexibilité que possible.

Le **Learning Registry** [Bienkowsky et al., 2012] est une infrastructure permettant aux enseignants et apprenants de découvrir et d'utiliser des ressources pédagogiques stockées dans différents systèmes internationaux, Américains en particulier. Learning Registry stocke des informations sociales telles que les tags, commentaires ou évaluations réalisées par les utilisateurs, en plus des traditionnelles métadonnées décrivant une ressource pédagogique. Ces données, appelées *paradata*, sont ensuite partagées dans une infrastructure commune à des fins d'agrégation et/ou d'analyse. Le schéma correspondant illustré par la Figure 3 est inspiré du schéma Activity Stream. L'élément racine est une collection d'activités caractérisées par (1) un acteur décrivant l'entité ou la personne ayant réalisé l'action, (2) un verbe traduisant le type d'activité, (3) l'objet sur lequel l'activité a été effectuée, (4) une liste d'objets liée à cet objet, et (5) une description textuelle et l'estampille horaire correspondant à la publication de l'activité.

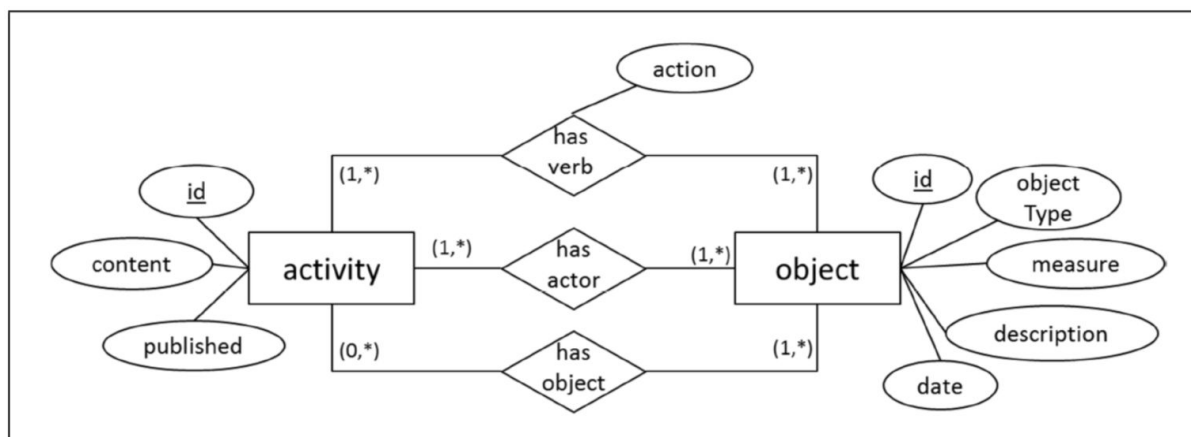


Figure 3. Schéma simplifié du Learning Registry [Niemann et al., 2012]

Dans l'exemple ci-dessous, la ressource localisée à <http://myboe.org/go/resource/110599> a été taggée avec 4 termes via le portail "Brokers of Expertise" par un éducateur le 6 juin 2015 à 9:12:00 heure locale.

```
{
  "doc_type": "resource_data",
  "doc_version": "0.50.0",
  "resource_data_type": "paradata",
  "active": true,
  "identity": {
    "submitter": "Brokers of Expertise",
    "submitter_type": "agent",
    "signer": "Brokers of Expertise"},
  "digital_signature": { ... },
  "resource_locator":
  "http://www.pbs.org/teachers/mathline/concepts/space2/activity3.sh
  tm",
  "payload_placement": "inline",
  "payload_schema": ["LR Paradata 1.0"],
  "resource_data": {
    "activity": {
      "actor": {
        "objectType": "educator",
        "description": ["Technology", "Science",
  "English-Language Arts"]},
      "verb": {
        "action": "tagged",
        "date": "20015-06-06T09:12:00",
        "context": "Brokers of Expertise Portal",
        "description": ["9 11", "911", "September 11th",
  "september 11"] },
      "object": {"id":
  "http://www.readwritethink.org/classroom-resources/lesson-plans/re
  el-995.html"},

```

"content": "La ressource localisée à
http://myboe.org/go/resource/110599 a été taggée avec 4 termes via
le portail Brokers of Expertise par un éducateur le 6 juin 2015 à
9:12:00 heure locale"
}}}

National Science Digital Library (NSDL²) propose des ressources digitales de qualité dans le domaine des sciences et technologies de l'ingénieur et des mathématiques à la communauté éducative. Les ressources sont décrites à l'aide de métadonnées LOM (e.g., domaine d'apprentissage, type d'audience et niveau) [LTSC, 2002], mais aussi par des données statistiques d'usage et des données renseignées par les utilisateurs du système (i.e., paradata telles que des évaluations ou des commentaires) [Blomer, 2012].

Le schéma de données décrivant un enregistrement NSDL est illustré par la Figure 4. Un enregistrement est identifié de manière unique, et renferme l'URL de la ressource pour lesquelles les paradata s'appliquent. L'élément le plus important décrivant un enregistrement NSDL est *usageDataSummary*, puisque celui-ci comprend différents types de données représentant différentes informations : le nombre de fois qu'une certaine action a été réalisée sur la ressource (e.g., commentée, lue, téléchargée, etc.) ; une valeur textuelle exprimant un commentaire ou un tag ; la valeur de l'évaluation moyenne de la ressource ; le nombre de « (un)like » d'une ressource ; le rang de la ressource au sein d'un ensemble de ressources. Cet élément contient également le laps de temps pendant lequel les données statistiques ont été collectées. Notons que d'autres informations peuvent également décrire un enregistrement, puisque n'importe quel élément supplémentaire peut être spécifié.

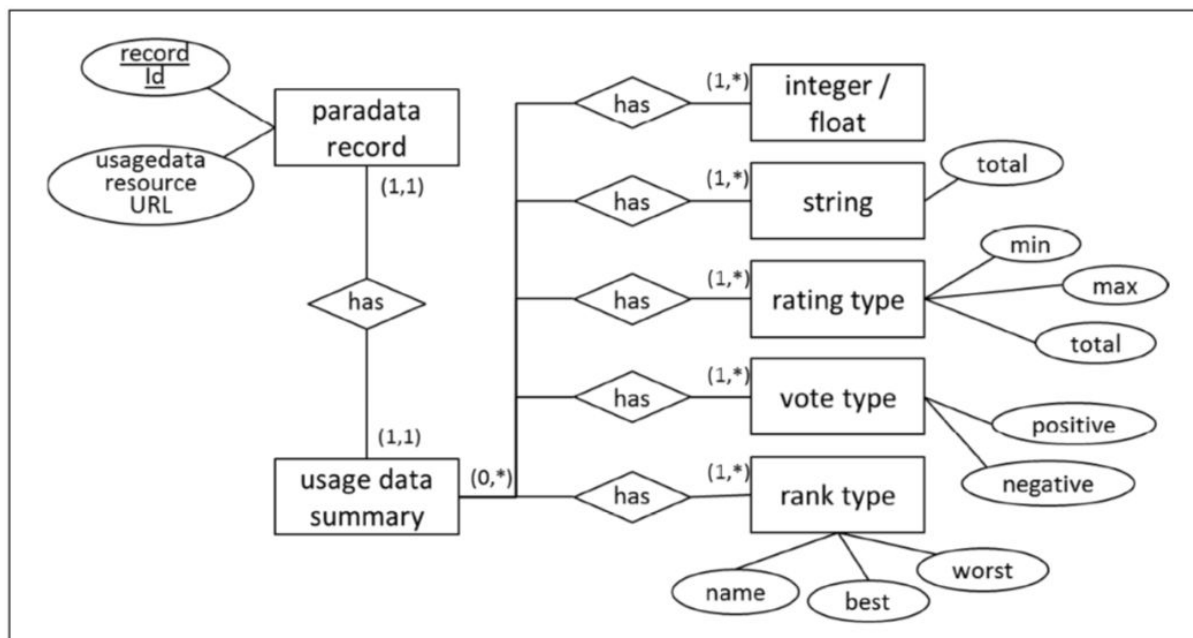


Figure 4. Schéma simplifié des paradata NSDL [Niemann et al., 2012]

Enfin, une autre initiative appelée Learning Context Data Model (LCDM) [Lukarov et al., 2014] a été développée dans le cadre du projet Learning Context³. Cette proposition se

² <http://www.nsd.org>

³ <http://learning-context.de/>

concentre sur des propriétés d'extensibilité et de mobilité, et adopte une approche centrée utilisateur pour représenter les données d'interaction issues d'environnements mobiles, de plates-formes web, etc. Le schéma illustré par la Figure 5 vise à représenter le contexte dans lequel se produisent les interactions avec une sémantique la plus riche possible. LCDM associe un utilisateur à un ensemble d'événements décrits par, entre autres informations, le nom de l'application à l'origine de l'évènement, le type de plate-forme utilisée (mobile, web, ordinateur de bureau), l'estampille horaire, le type d'action réalisée par l'utilisateur, la catégorie de l'évènement (privé, professionnel, ou académique), et l'entité sur laquelle s'est produit l'évènement (par exemple le titre d'une ressource éducative).

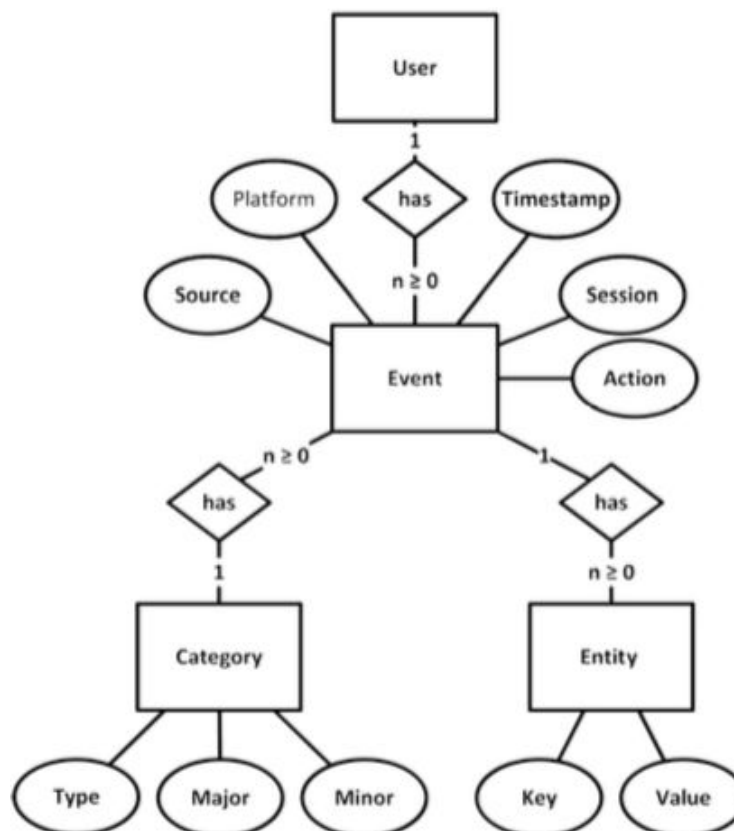


Figure 5. Schéma LCDM [Muslim et al., 2016]

Au-delà de ces initiatives internationales, des standards voient le jour et sont intégrés dans les outils existants ou en cours de développement. Deux principaux standards sont largement utilisés dans les outils de collecte et d'analyse de traces : xAPI et IMS Caliper.

Le tableau ci-dessous expose les diverses initiatives propriétaires et standardisées décrites dans cette section, et synthétise leurs capacités selon différents critères :

- la prise en compte d'un modèle détaillé de l'utilisateur ;
- la description de l'activité réalisée ;
- la collecte d'informations relatives au contexte d'apprentissage dans lequel l'activité a pris place ;
- la flexibilité réfère à la capacité du modèle à prendre en considération des applications et activités hétérogènes ;
- le degré d'expressivité représente le niveau de détails avec lequel le modèle de traces représente les informations collectées ;

- l'extensibilité dénote la capacité du modèle à être étendu pour satisfaire de nouveaux besoins.

	Modèle de l'utilisateur	Modèle de l'activité	Modèle du contexte d'apprentissage	Flexibilité	Expressivité	Extensibilité
Attention. XML						
CAM						
Activity Stream						
Learning Registry						
NSDL paradata						
LCDM						
IMS Caliper						
xAPI						
Légende						
	Non supporté					
	Partiellement supporté					
	Supporté					

Comme on peut le constater la préparation du jeu de données est une étape incontournable des Learning Analytics. Il existe un certain nombre de propositions concernant le type d'information (Modèle de l'utilisateur, modèle d'activité, ...) à modéliser. Ces propositions servent souvent de guide dans les prototypes construits ad hoc afin de faire une sélection des informations importantes. En parallèle avec ces propositions, les outils informatiques nécessitent de prendre en compte un format de stockage des données. Pour cela, plusieurs options sont possibles mais le plus souvent l'utilisateur doit convertir ses données dans un format un minimum standardisé. Des outils comme kTBS ou UTL permettent l'importation de traces dans différents formats (CSV, XML, JSON), d'autres comme UnderTracks se basent sur des systèmes de gestion de bases de données classiques (SQL). Les logiciels de statistiques plus généralistes fonctionnent aussi avec l'importation de formats standards.

Enfin, des initiatives autour de la norme xAPI, exprimée en JSON, ont été proposées comme CSVtoXAPI qui facilitent la conversion vers xAPI. Toutefois, chacun de ces outils

nécessite l'identification des informations et la mise en forme dans les formats standards (CSV, XML, JSON). Pour aider à la préparation des données, le logiciel TALEND (décrit dans le document sur les outils professionnels) par exemple, aide à l'extraction et la conversion des informations avec des processus plus ou moins automatisés.

Analyser les jeux de traces

De nombreux travaux se concentrent sur l'étape d'analyse à proprement parler en suivant des approches différentes et en visant des publics différents.

Outils généralistes de statistique

Une première approche consiste à utiliser des **outils généralistes de statistique** (R, SAS, Stata, SPSS, SPAD, etc.) pour la mise en oeuvre de méthodes couramment utilisées dans le champs des sciences humaines, autrement dit des outils non spécifiques de l'analyse de données de type traces. Ces outils permettent principalement d'analyser des données tabulaires, mais certains d'entre eux donnent également la possibilité d'analyser des données séquentielles, ce qui requiert alors l'utilisation de méthodes spécifiques de ce format. Ces outils se distinguent d'outils plus orientés, tels les logiciels conçus prioritairement pour les données issues de sondages (ergonomie facilitant la lecture des variables à choix multiples, l'utilisation de poids de sondage, de sous-populations, etc.), ou encore des logiciels dédiés à l'analyse statistique de corpus textuels (basés sur la fréquence de cooccurrence de mots ou sur la fréquence de champs sémantiques), etc.

Avec les outils généralistes, pour les analyses statistiques classiques les traces recueillies doivent se présenter au format « standard » de ces outils, à savoir sous forme de tables de données croisant en ligne des individus statistiques et en colonne des variables (aussi appelées descripteurs, caractéristiques). Dès lors, que ce soit directement si le format convient ou après transformation (pré-traitements avant l'importation des données), ces outils permettent d'appliquer un ensemble de méthodes de traitement (recodages, création de nouvelles variables, etc.) et de méthodes d'analyse.

Parmi les méthodes d'analyse, il s'agit principalement :

- De **décrire** les caractéristiques de l'ensemble des individus statistiques ou de sous-populations en résumant chaque variable isolément (tris à plat, moyenne, médiane...), deux variables simultanément (corrélations, tri-croisés) ou encore 3 variables et plus simultanément. Dans ce dernier cas, les analyses factorielles de type Analyse en Composantes Principales (variables quantitatives) ou Analyse Factorielle des Correspondances (variables qualitatives) visent à restituer le meilleur résumé, notamment visuel, des proximités entre variables d'une part et entre individus d'autre part.
- De **comparer** des sous-populations : tests de comparaison de moyennes, de comparaison de distributions, etc.
- De **classer** les individus au regard d'un ensemble de variables par classification supervisée (Analyse Factorielle Discriminante, arbres de décision...), le principe étant de définir à partir d'un échantillon d'apprentissage les règles discriminant au

mieux les individus de sorte de pouvoir classer de nouveaux individus, ou bien de regrouper les individus par classification non supervisée (Classification Ascendante Hiérarchique, méthode des centres mobiles...) afin de constituer des groupes homogènes, appelés profils ou classes, à l'aide d'une distance calculée entre individus (distance tenant compte des différentes variables de l'analyse). Une fois les profils obtenus, l'interprétation du résultat passe par une synthèse, produite par l'analyste, des caractéristiques qui confèrent une homogénéité interne aux profils et inversement une hétérogénéité entre profils.

- De **modéliser** une variable au regard d'un ensemble d'autres variables. Les modèles de régression se déclinent en différentes familles de modèles répondant à différents contextes d'analyse (selon le type et la distribution des variables). Parmi les modèles courants on peut citer l'Analyse de la Variance (ANOVA), qui vise à déterminer si une ou plusieurs variables qualitatives (on parle souvent de "facteurs" ou "conditions" dans le cas d'expérimentations) sont influentes sur une variable quantitative.

Parmi les méthodes visant à modéliser une caractéristique on peut distinguer deux objectifs, ces deux objectifs n'étant pas exclusifs l'un de l'autre :

- Expliquer : les modèles de régression linéaire multiple, par exemple, permettent de déterminer, pour une caractéristique cible (à expliquer), quelles sont les caractéristiques explicatives ou autrement dit influentes sur les valeurs de la caractéristique cible et celles qui ne le sont pas, d'évaluer l'effet d'une caractéristique toutes autres caractéristiques étant égales par ailleurs et de définir le poids de chacune des caractéristiques influentes du modèle. Pour aller plus loin, afin de tenir compte du contexte dans lequel se trouvent les individus (par exemple la classe, l'établissement ou un autre contexte d'apprentissage), les modèles multiniveaux permettent d'intégrer à la fois des variables descriptives du contexte (niveau 2) et des variables descriptives des individus (niveau 1) afin de modéliser une variable mesurée sur les individus (niveau 1).
- Prédire : les arbres de décision, parmi les méthodes d'apprentissage supervisé, donnent hiérarchiquement les descripteurs les plus discriminants d'une variable cible, le résultat étant produit sous forme schématique d'arborescence facilitant la lecture du résultat. Ces méthodes permettent notamment de prévoir la valeur de la caractéristique cible pour un nouvel individu dont on connaît les valeurs des différents descripteurs.

Lorsque les données recueillies par les outils d'apprentissages retracent l'activité **séquentiellement** c'est-à-dire sous la forme d'une **suite ordonnée d'actions**, certains outils généralistes de statistique permettent de déployer des méthodes, en particulier les Méthodes d'Appariement Optimal [Lesbard et al., 2006], visant à classer les séquences et par extension les individus, en fonction de la suite d'éléments qui composent les séquences. Il s'agit d'une méthode de classification non supervisée utilisant une distance entre séquences deux à deux. Cette distance est établie en paramétrant le coût d'une insertion, d'une suppression et d'une substitution afin de calculer pour chaque paire de séquences le coût total des modifications nécessaires pour rendre les deux séquences identiques. Une fois les profils obtenus, l'homogénéité des séquences regroupées dans un profil peut ensuite

être visualisée à l'aide de chronogrammes (ou "Time Line") et décrite au regard des actions les plus fréquentes, des patterns d'actions, etc.

Le format séquentiel des données n'exclut cependant pas le passage vers des données tabulaires calculées à partir des séquences (on parlera alors de données agrégées) afin d'obtenir des descripteurs de celles-ci (ex : nombre d'actions d'un type donné, durée des séquences, etc.) puis l'application des méthodes statistiques classiques énoncées ci-dessus.

Concernant la démarche globale de l'analyste, elle peut avoir une visée exploratoire et/ou confirmatoire, à savoir une découverte préliminaire de la distribution des données pour procéder ensuite à leur description multidimensionnelle (par exemple la recherche de profils sans modèle *a priori*), et/ou une mise à l'épreuve de modèles théoriques ou d'hypothèses de recherche. Pour une partie des méthodes statistiques elles requièrent une mise en oeuvre par des spécialistes.

Pragmatiquement, même si un ensemble de calculs peuvent être opérés à l'aide de scripts, cette approche se distingue de celle utilisant des algorithmes automatiques car l'utilisation des outils généralistes implique une progression pas à pas dans l'analyse. En effet les résultats obtenus au fil de l'analyse ont une incidence sur les choix suivants de méthodes, choix de paramétrages et de spécification des modèles. Les deux approches répondent à des objectifs différents.

Algorithmes automatiques

Un autre approche consiste à utiliser des **algorithmes automatiques** d'analyse de traces afin d'obtenir des informations qui sont soit exploitées automatiquement par des systèmes, soit affichées aux différents acteurs.

Ainsi, Laalys [Muratet et al., 2016] est un outil d'analyse reposant sur un réseau de Pétri qui permet d'associer des étiquettes pédagogiques aux actions des élèves et calcule un score à partir de ces étiquettes. Ces dernières renseignent les enseignants sur le comportement d'un élève. Le principe de cet algorithme d'étiquetage est de fournir des informations sémantiques en caractérisant les écarts détectés entre la résolution de l'élève et celles préconisées par les enseignants. Le résultat de l'analyse est alors affiché aux enseignants via une interface graphique. Cet outil est présenté en détail dans la section 4.

Autre exemple, T-Store [Zarka et al., 2013] est un système de gestion des traces (SBT) qui gère le stockage, la transformation et l'exploitation des traces collectées par des applications externes. Pour exploiter les traces, des transformations sont utilisées. T-Store fournit des fonctions de transformation prédéfinies ainsi qu'une transformation personnalisée basée sur les automates à états finis. Ces transformations permettent d'extraire des connaissances qui sont ensuite exploitées pour fournir de l'assistance aux utilisateurs finaux, et notamment des recommandations.

D'autres travaux utilisent des algorithmes automatiques, mais en laissant l'analyste intervenir dans le paramétrage de l'algorithme. Ainsi, DisKit [Fuchs, 2018] permet d'extraire

des connaissances présentes dans les traces *via* l'utilisation de l'algorithme dmt4sp [Mannila et al., 1997]. Cet outil permet d'extraire des motifs décrivant des séries d'événements selon des contraintes, pouvant être ajustées par l'analyste, concernant le nombre minimum d'occurrences des motifs, sur leur taille...

Langage dédié : du langage informatique à la langue naturelle

Une autre approche consiste à **utiliser un langage dédié** à la manipulation des traces.

Ainsi, l'environnement UTL [Iksal, 2012][Iksal, 2011] (Usage Tracking Language) a été conçu pour la conception et l'opérationnalisation d'indicateurs prescrits. Il est nécessaire dans ce contexte de savoir ce que l'on souhaite observer et pour quelle raison. Ensuite, cet environnement permet la description des données d'analyse ainsi que de leurs règles de calcul dans un format indépendant de toute plateforme d'apprentissage. Cette solution facilite la réutilisation des descriptions dès lors que les éléments sur lesquels sont basées les analyses existent dans les traces importées. UTL est composé d'un éditeur web pour les données, d'un calculateur pour l'opérationnalisation ainsi que de connecteurs permettant l'élaboration de tableaux de bord.

KTBS (*kernel for Trace-Based Systems*) [Champin et al., 2013] est une implémentation de référence open-source d'un Système à base de Traces Modélisées (SBTm). La notion centrale des SBTm est celle de trace modélisée, définie comme une liste d'éléments observés appelés obsels. Chaque obsel est décrit par un type, un ensemble d'attributs, et deux estampilles temporelles début et fin, délimitant l'intervalle durant lequel cet obsel a pu être observé. Chaque trace est associée à un modèle de traces, qui spécifie les types d'obsels que la trace peut contenir, ainsi que les attributs de chaque type d'obsels. Ainsi, le modèle de traces permet d'explicitier la structure et la sémantique sous-jacente d'une trace. Cette connaissance est capitalisable, puisque plusieurs traces décrivant des activités similaires peuvent faire référence au même modèle. Le KTBS utilise le modèle de données RDF⁴, qui offre la flexibilité nécessaire pour représenter les traces modélisées selon divers modèles de traces. KTBS fournit un ensemble d'opérateurs de transformation, depuis de simples filtres jusqu'à des ré-écritures complexes spécifiées en SPARQL⁵. Il permet également la définition d'opérateurs personnalisés. Tous ces opérateurs sont à coder en utilisant le langage RDF.

Une autre approche consiste à exploiter la langue naturelle pour interroger les données et ainsi permettre à des analyses non-informaticiens d'interroger de Système à Base de Traces. Ainsi, SPARE-LNC (pour SPARql REquest en Langage Naturel Contrôlé) [Kong et al., 2015] est un langage dont l'objectif est de proposer une alternative au SPARQL pour interroger les traces stockées dans le système à base de traces KTBS. Ce langage est guidé par une grammaire algébrique, ayant comme base soit la langue française, soit la langue anglaise. L'entrée de l'utilisateur est analysée dans son intégralité via l'ensemble de règles composant la grammaire. L'ordre des règles n'est pas absolu, permettant certaines libertés. On peut différencier dans la définition de la grammaire deux groupes de règles. Le premier

⁴ <https://www.w3.org/RDF/>

⁵ <https://www.w3.org/TR/rdf-sparql-query/>

groupe permet la création de phrases exprimant des conditions sur des éléments à récupérer dans les traces. Le deuxième permettant la définition de phrases gérant les éléments récupérés, par exemple en opérant des calculs ou des sélections sur ce qui est récupéré.

Le langage utilisé pour requêter la base est ainsi composé d'un ensemble de ces phrases qui satisfont la grammaire proposée. Chacune de ces phrases correspond ainsi à une sous-requête du langage, formant un texte décrivant les données à récupérer en énonçant un ensemble de contraintes et un ensemble d'actions à réaliser.

Cet ensemble de phrases est ensuite traduit automatiquement en SPARQL pour interroger la base de traces (cf. figure 6 ci-dessous).

```
Je cherche à récupérer les obsels de type SimpleObsel et ayant un attribut
de valeur 'Erik' suivi par un obsel ayant un attribut de valeur "someone
else".

prefix : <http://liris.cnrs.fr/silex/2009/ktbs#>
prefix ns1: <http://liris.cnrs.fr/silex/2011/simple-trace-model/>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
prefix xml: <http://www.w3.org/XML/1998/namespace>
prefix xsd: <http://www.w3.org/2001/XMLSchema#>

prefix base: <http://liris.cnrs.fr/silex/2011/simple-trace-model/>
prefix model: <http://liris.cnrs.fr/silex/2011/simple-trace-model/>

SELECT DISTINCT ?sobs0 ?pobs0 ?cobs0
WHERE
{
  ?sobs0 ?pobs0 ?cobs0 .
  ?sobs0 :hasEnd ?dateEndobs0 .
  ?sobs0 :hasBegin ?dateBeginobs0 .
  ?sobs0 rdf:type base:SimpleObsel .
  FILTER( ?cobs0 1 = 'Erik' )
```

Figure 6. Traduction automatique en SPARQL

Outils graphiques

Une dernière approche consiste à fournir des **outils de manipulations graphiques** des traces. Ainsi, DDART [Michel, et al, 2017] permet aux étudiants et aux enseignants utilisant Moodle de combiner des traces hétérogènes, puis *via* une interface graphique de concevoir des indicateurs en choisissant les éléments de traces, le type de calcul à effectuer, les paramètres des calculs et le format de visualisation des indicateurs. Un calcul dynamique et un affichage en temps réel permet facilement à l'utilisateur d'adapter la conception de l'indicateur pour atteindre la forme voulue. D'autres travaux se concentrent sur la proposition de tableaux de bord dynamiques.

SBT-IM (Système à Base de Traces pour le calcul d'Indicateurs sur la plateforme Moodle) [Djouad et al., 2011] est un Système à base de Traces spécifique dédié à la définition d'indicateurs d'activité collaboratifs et individuels dans les activités de la plateforme collaborative Moodle⁶. SBT-IM permet de définir un indicateur et de choisir une visualisation

⁶ https://moodle.org/?lang=fr_fr

en présentant à l'auteur un système de renseignement d'informations progressif, des informations générales aux informations détaillées sur le calcul de l'indicateur en cours de définition. Le parcours des traces se fait via une suite de tableaux affichant les données filtrées par opérations successives.

DDART [Michel, et al, 2017] est un outil qui peut s'intégrer à un ENT conçu avec Moodle pour offrir, en complément des outils de gestion du travail collaboratif (forum, chat, wiki, ...) de Moodle, deux outils de planification et de suivi de l'activité des élèves et du groupe : le reporting tool permet de spécifier différents objectifs à atteindre, de décrire comment l'activité se réalise ou de juger la qualité de la réalisation intrinsèque ; le tableau de bord (*dashboard*) offre une possibilité de création d'indicateurs sur la base des traces de l'activité collectées manuellement à partir du *reporting tool* et des traces de l'activité enregistrées automatiquement s'il y a usage de Moodle dans la formation. L'interface de conception des indicateurs permet, par des *drags and drops* entre les zones "paramètre" et "design d'indicateur", de spécifier : les entités (les éléments sur lesquels se font les calculs) ; les types de données (fréquence, intervalle de temps, contenu, description) liées aux entités ; les types de calculs ; et enfin les types de visualisations. La spécification de tout nouvel élément dans la zone "indicator design" provoque un affichage direct du visuel de l'indicateur. Ce calcul dynamique permet à l'utilisateur d'adapter facilement la conception de l'indicateur pour atteindre la forme voulue. Le tableau de bord offre une interface de visualisation de l'ensemble des indicateurs créés. Cet outil est présenté en détail dans la section 4.

Abstract [Georgeon et al., 2012] est une application web proposant des outils graphiques de manipulation des traces pour l'analyse de l'activité humaine en temps réel. L'objectif est de pouvoir mener une analyse de traces d'activité pour la modélisation cognitive de l'utilisateur. L'outil propose différentes fonctionnalités : un éditeur d'ontologie pour spécifier les modèles des différentes traces, un éditeur de transformation pour spécifier différentes règles de transformations applicables aux traces, un moteur de transformation, un système de visualisation des traces et du résultat des transformation et enfin un outil de requête pour recherche des occurrences de schéma dans les traces.

KTBS4LA (KTBS for Learning Analytics) est une surcouche de KTBS permettant l'interprétation des traces et la manipulation des concepts manipulés dans le KTBS, sans nécessité de connaissance en langage de programmation. Cette application web capitalise les travaux proposés par Samotrices⁷ et SamotraceMe [Derbel et al., 2015] sur la visualisation de traces multi-vues, multi-échelles et multi-sources, réexploitent la collection de composants web de Taaabs⁸ pour manipuler et visualiser graphiquement les traces contenues dans un KTBS et intègre le langage SPARE-LNC [Kong et al., 2015]. Pour analyser les traces d'interaction issues d'un EIAH, un utilisateur de KTBS4LA procédera en plusieurs étapes. Il doit tout d'abord importer les traces issues de cet EIAH dans KTBS4LA, ce qui lui permet également de définir le modèle des traces pour une situation correspondante à l'usage de cet EIAH. Il peut ensuite explorer les traces importées à l'aide de différents outils de visualisation, et créer de nouvelles traces modélisées permettant de

⁷ Site du projet SAMOTRACES : <http://sourceforge.net/projects/samotrices/>

⁸ Site du projet TAAABS : <https://projet.liris.cnrs.fr/sbt-dev/tbs/doku.php/tools:taaabs>

mieux comprendre l'activité des apprenants. Cet outil est présenté en détail dans la section 4.

D3KODE [Champalle et al., 2016] (Define, Discover and Disseminate Knowledge from Observation to Develop Expertise) est une plateforme web bâtie sur la base du kTBS dans un but de réutilisation et de partage de connaissances d'analyse de traces numériques. L'interface de D3KODE s'adresse à des utilisateurs non informaticiens et a donc été conçue de manière à faciliter l'analyse de traces numériques en dégageant l'utilisateur des notions techniques du kTBS. Les fonctionnalités du prototype permettent aux utilisateurs d'importer des données de bas niveaux et de les transformer en informations de plus hauts niveaux. Le résultat est présenté sous la forme d'une synthèse visuelle sur plusieurs niveaux d'abstraction où chaque "observation" d'un niveau N est reliée à ses origines dans le niveau N-1. Les niveaux sont construits *via* des règles créées au travers d'une interface dédiée. Les règles créées sont réutilisables et donc partageables entre utilisateurs. La synthèse visuelle est interactive, et permet à un analyste d'accéder aux informations de chaque niveau, règle et observation en cliquant sur le points qui l'intéresse.

UNDERTRACKS [Bouhineau et al., 2013] est un outil d'assistance à la création de processus d'analyse qui permet de guider les choix d'opérateurs en fonction de vues sur les données. Il permet l'import et le stockage des traces, la gestion des opérateurs d'analyse ainsi que la construction visuelle des processus d'analyse et la représentation graphique des résultats. Toutefois, les données produites ne sont pas persistantes, et nécessitent donc d'être re-calculées chaque fois que l'on souhaite les ré-exploiter.

Visualiser les résultats de l'analyse

Présenté en introduction de ce rapport, le cycle d'analyse de traces passe par plusieurs étapes : collecte des traces, analyse, visualisation des traces et des résultats produits par l'analyse [Fayyad et al., 1996] [Clow, 2012] [Stamper et al., 2011].

La majeure partie des outils cités dans cet état de l'art permettent de parcourir une grande partie de ce cycle. Tous ne possèdent cependant pas une « visualisation graphique » qui se prête à l'analyse des traces et/ou à la compréhension des comportements des apprenants dans les EIAHs.

Les outils retenus dans cette section, possèdent une visualisation graphique des traces "intégrée" (et non externe) : Abstract, D3kode, DDART, kTBS4LA, Travis, SBT-IM, Tatiana, UnderTracks, Lab4CE, Emoda et Transmute. Nous avons donc fait le choix d'exclure de cette section tous les outils qui ne possèdent pas d'aspect visualisation graphique autre que textuel tels par exemple Laalys ou UTL.

En terme de visualisation graphique, parmi les outils cités dans cette section, il est possible de distinguer deux catégories d'applications :

- les outils d'exploration "généralistes" de traces numériques non exclusivement dédiés aux LA, mais dont l'utilisation est aussi possible et avérée dans d'autres contextes;

- les outils de types “tableaux de bord”, conçus spécifiquement pour les traces numériques provenant d'EIAHs. Ils possèdent un panel d'indicateurs graphiques élaborés pour “observer” l'activité suivant plusieurs orientations. Ils ciblent les apprenants et/ou les enseignants/tuteurs.

Outils d'exploration généralistes

Parmi les formes de visualisation graphique les plus représentatives et les plus courantes des outils “généralistes”, la “Time Line”⁹ est la plus récurrente. Il en existe plusieurs variantes parfois associées avec d'autres visualisations graphiques plus classiques telles des histogrammes ou des camemberts :

- Abstract [Georgeon et al., 2012] propose une visualisation unique d'une seule trace numérique dans une optique d'analyse et de découverte de l'activité. La trace représentée, est soit une trace première (avant transformation) soit une trace abstraite (transformée) avec des liens “origines” entre ses observés. Abstract est tourné en direction de l'analyste et n'est pas dédiée exclusivement aux traces d'EIAHs.

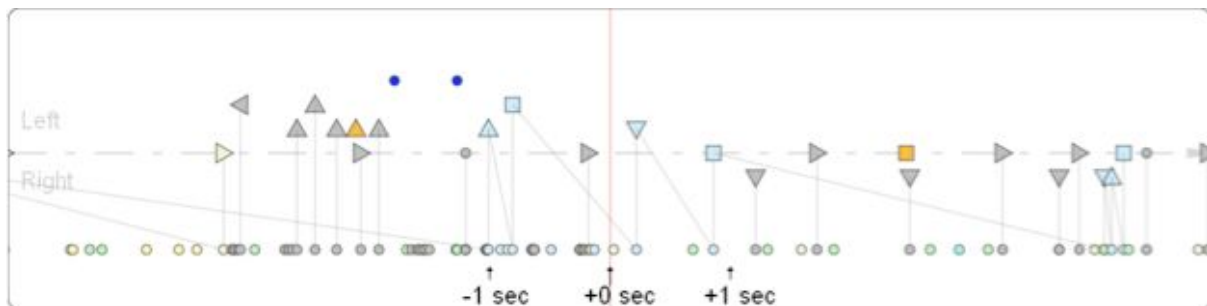


Figure 7. Visualisation d'Abstract

- Dans la lignée d'Abstract, D3KODE possède une « Time Line » plus riche [Champalle et al., 2016]. La représentation est interactive et plusieurs niveaux d'abstraction peuvent être représentés. D3KODE est tourné sur la capitalisation et le partage des connaissances d'analyses; la traçabilité et la réutilisation des connaissances de transformations (abstractions) mobilisées est plus développée. Cette traçabilité permet aux utilisateurs de comprendre comment les niveaux d'abstractions et leurs observés sont construits.
- TRANSMUTE [Barazzutti et al., 2016] est une interface graphique interactive et personnalisable conçue pour assister l'interprétation de traces et la découverte de connaissances. La visualisation des traces se présente sous la forme d'une Time Line assez classique (dans la lignée d'Abstract et de D3KODE) permettant de visualiser très rapidement la trace « en cours » et la trace « transformée » par l'analyste. La figure suivante présente l'interface de transmute couplée à Diskit, une application orientée fouille de données.

⁹ Une time line est une représentation symbolique des événements sur un axe temporel horizontal permettant d'explorer la dimension temporelle des données sur une période de temps qui peut être/ou non paramétrée (mn, heure, jour, mois,...).



Figure 8. Interface de Transmute

- ktBS4LA est une plateforme web d'analyse de trace dont l'objectif est de faciliter la manipulation graphique des données collectées ainsi que leur visualisation rapide. La vue principale des données est une « Time Line » dont l'organisation graphique des observés en « colonne » permet une vue « en profondeur » de l'activité tout en conservant un aperçu global sur une période de temps importante. Pour compléter l'analyse, ktBS4LA propose aussi l'emploi d'autres indicateurs tel des histogrammes et des camemberts [Casado et al., 2017].
- Tatiana [Dyke et al., 2010], outil d'analyse graphique de traces conçus pour des chercheurs en sociologie, exploite aussi le principe de Time Line pour présenter les interactions médiatisées et assister leur analyse.

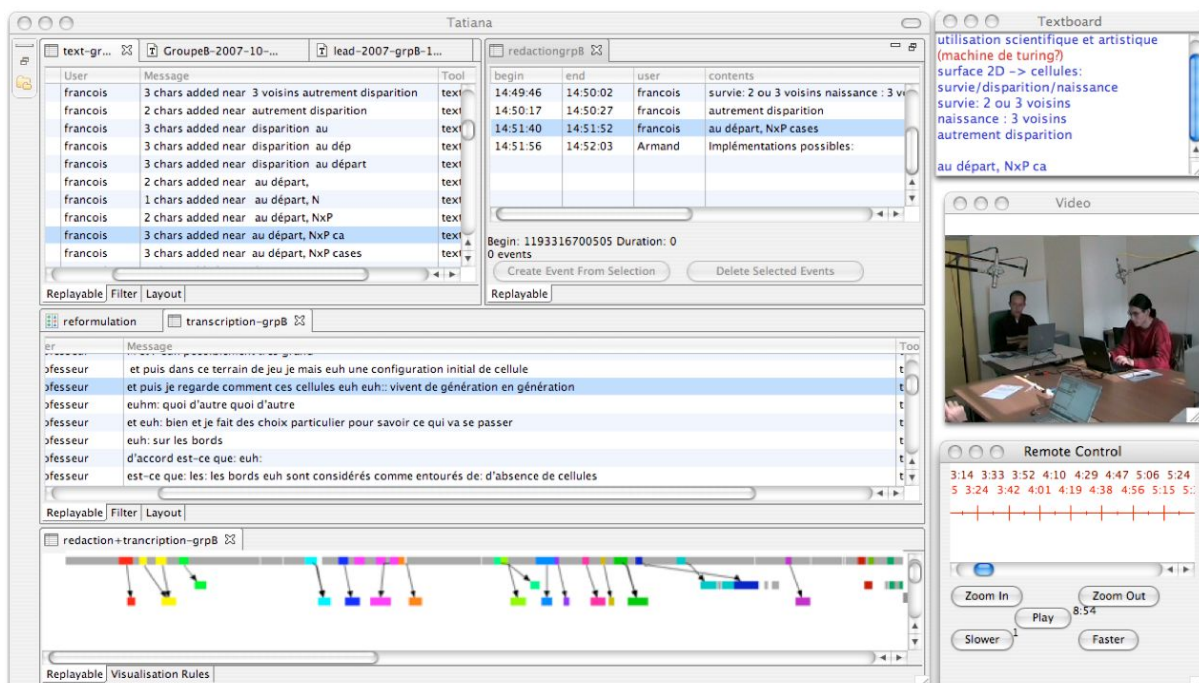


Figure 9. Chronotime dans l'outil Tatiana

- UNDERTRACKS [Bouhineau et al., 2013] est un outils d'analyse "généraliste" basé sur la plateforme "orange". Dédié à l'origine à la capitalisation de processus d'analyse de traces d'apprenants, il peut être utilisé pour d'autres contextes. Undertracks propose un panel extensible de visualisation graphique, dont un pattern de type Time Line offrant notamment la possibilité de comparer plusieurs traces d'activités.

Etudiant 654



Etudiant 656



Etudiant 673



Figure 10. Comparaison de plusieurs trace dans UnderTracks

Outils type "tableau de bord" dédiés aux EIAHs

Les outils de types "tableaux de bord" sont dédiés à l'analyse de traces numériques provenant de plateforme d'apprentissage spécifiques. Ils possèdent un panel d'indicateurs graphiques élaborés pour "observer" l'activité suivant plusieurs facettes. Les utilisateurs ciblent peuvent être les apprenants, dans un but réflexif et/ou les tuteurs dans un but de compréhension et/ou de régulation de l'activité d'apprentissage.

- Travis [May et al., 2011] est un outil d'analyse de forums dédié à la plateforme Moodle. Il utilise les traces numériques de Moodle afin de proposer à l'enseignant une vue d'ensemble interactive des échanges et consultations du forum par les élèves. Travis proposent plusieurs indicateurs :
 - une time line conçues pour faciliter l'exploration des échanges du forum avec des bulles plus ou moins importantes selon le nombre de messages échangés dans un fil de discussion.
 - plusieurs indicateurs statistiques pour visualiser l'activité individuelle des étudiants sur le forum

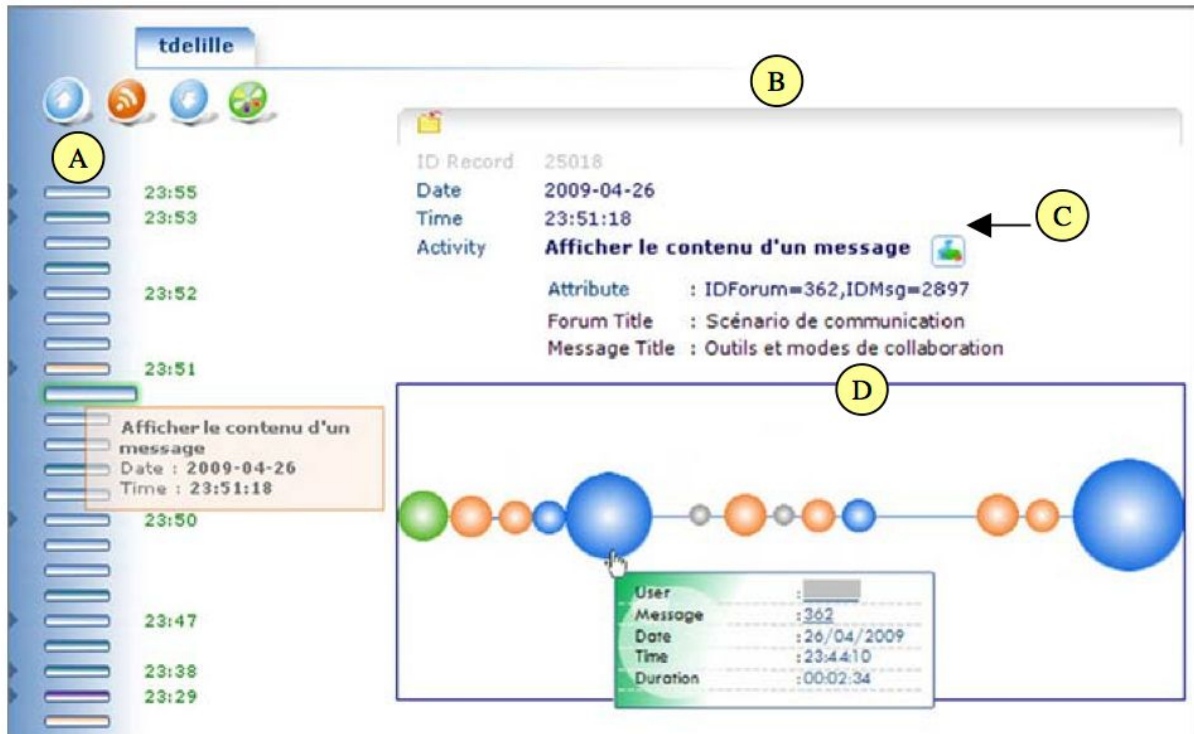


Figure 11. Interface de Travis et la time line des échanges dans un forum

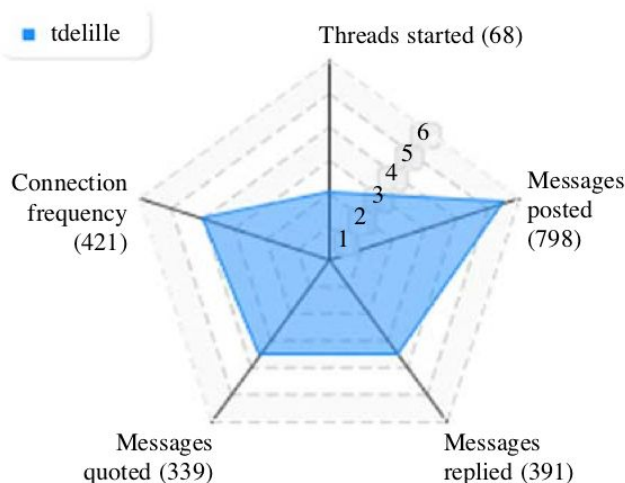


Figure 12. Quelques exemples d'indicateurs dans Travis

- SBT-IM [Djouad et al., 2011] permet de collecter des traces à partir des plateformes d'apprentissages Moodle dans le but de créer des indicateurs pour analyser les traces d'activités des étudiants sur la plateforme. SBT-IM permet la création et la réutilisation d'indicateurs ainsi que plusieurs visualisations graphiques interactives de l'activité. L'activité des étudiants peut par exemple être visualisée chronologiquement mais aussi sous forme d'indicateurs plus classiques de type histogramme ou camembert.

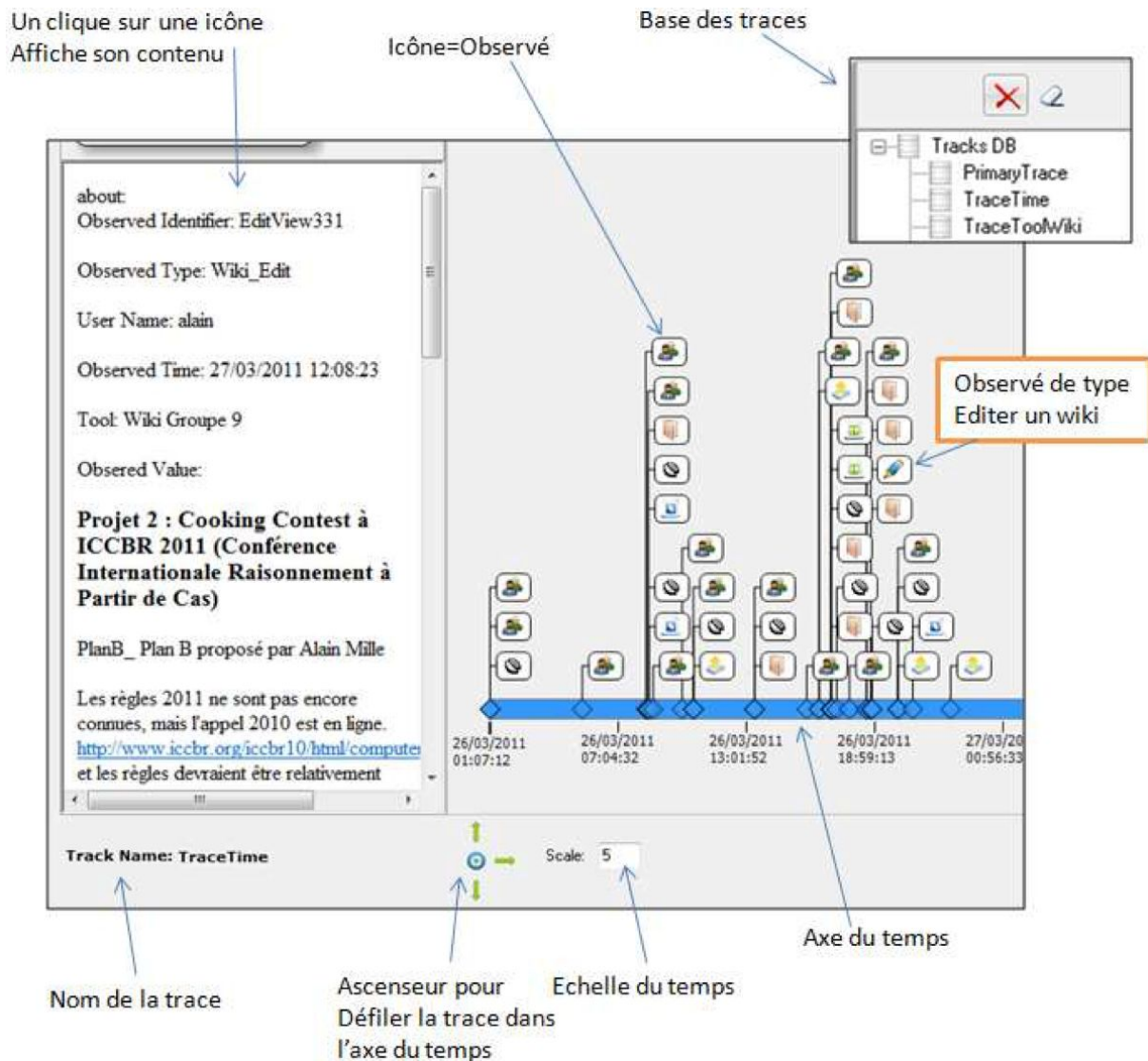


Figure 13. Visualisation dans SBT-IM

- LAB4CE [Broisin et al., 2017a] [Broisin et al., 2017b] est une plateforme web munie d'un ensemble de fonctionnalités support à l'apprentissage et d'un outil de visualisation en direction du tuteur et des apprenants. Lab4ce propose plusieurs types de visualisations graphiques principalement réflexif :
 - **L'outil de comparaison sociale** : un ensemble de barres de progression qui reflète le niveau de performance des apprenants par un code couleur

- **Réflexion a posteriori de l'activité d'apprentissage** : une time line permettant aux apprenants d'analyser en détail leurs propres actions ainsi que celles de leurs pairs. Pour chaque ressource sélectionnée, une chronologie des instructions exécutées est représentée. Chaque nœud de la chronologie représente une instruction ; celui-ci est coloré selon sa justesse technique, tandis que le détail de l'instruction apparaît dans un simili-terminal lorsque le curseur est positionné sur le nœud correspondant.
- **Analyse des stratégies mises en œuvre** : un indicateur de type time line permettant aux apprenants de visualiser l'évolution des stratégies qu'ils mettent en œuvre tout au long de la réalisation des activités d'apprentissage.
- EMODA [Ez-Zaouia et al., 2017] est un tableau de bord conçu pour aider les enseignants engagés dans des formations en ligne pour l'apprentissage des langues. La particularité de l'approche EMODA réside dans son ambition de détecter les émotions des apprenants pour faciliter le lien entre les enseignants et les apprenants à distance.
EMODA adopte une approche multimodale et considère quatre sources de données : audio, vidéo, self-report, et traces d'interaction.
Les données audio et vidéo correspondent aux échanges de communication enregistrés lors des séances. Les données de self-report sont renseignées par l'apprenant avant et après la séance. Les traces d'interactions sont utilisées pour qualifier les actions liées à des émotions particulières au cours de l'activité.
EMODA propose des résultats sous forme d'indicateurs discrets, bi ou multidimensionnels. La visualisation des émotions est transmise à l'enseignant au travers d'un tableau de bord avec différents formats de visualisation : histogramme, émoticône, courbes d'évolution temporelle, images significatives de la séance. (Plus de précision en section 4).
- DDART [Michel, et al, 2017] pour Dynamic Dashboard Based on Activity and Reporting Traces est en même temps un outil de reporting et de monitoring. L'approche de DDART est centrée apprenant dans un contexte Project-Based Learning (PBL). L'objectif de DDART est d'aider les apprenants à collecter, analyser et visualiser les traces significatives de leurs activités par eux-mêmes (DDART est décrit plus en détail en section 4).
DDART peut s'intégrer à un ENT conçu avec Moodle. L'intégration de DDART offre, en complément des outils de gestion du travail collaboratif (forum, chat, wiki, ...) de Moodle, deux outils de planification et de suivi de l'activité des élèves et du groupe :
 - ❑ le *reporting tool* permet de spécifier différents objectifs à atteindre, de décrire comment l'activité se réalise ou de juger la qualité de la réalisation intrinsèque. Les élèves peuvent ainsi collecter des éléments d'information sur leur activité, gérer leurs rapports d'activité, les modifier *a posteriori* ou faire des commentaires généraux sur les contextes des activités. Les élèves peuvent également lire et commenter les rapports des autres élèves du

groupe, ce qui est très utile pour que l'élève se situe par rapport au groupe et puisse aussi prendre du recul sur son activité ;

- ❑ le *dashboard* offre une vue globale sur des indicateurs de suivi des activités rapportées et des activités réalisées avec les outils de l'ENT. Il permet ainsi de suivre qui travaille sur quoi, combien de temps, quel est le jugement des uns et des autres... Ces indicateurs sont créés dynamiquement par les élèves ou l'enseignant en fonction des besoins du projet pédagogique.

3 - Synthèse des outils étudiés

Une analyse détaillée (cf. Figure 13) des outils que nous avons étudiés, montre que les outils sont essentiellement des logiciels et que l'analyse des données et leur visualisation représente plus de 50% des outils disponibles. En ce qui concerne le public visé, c'est en priorité les enseignants et les analystes, cela s'explique de part le fait que les analyses sont souvent réalisées en fonction d'un besoin des enseignants d'observer le déroulement de leur situation d'apprentissage. Pour les analystes, cela est dû principalement au fait que de nombreux outils ont été conçus dans le cadre de la recherche et de travaux de chercheurs en analyse de données. Ces analyses souvent complexes, nécessitent au préalable un travail de la part de spécialistes des données. Enfin, en ce qui concerne les formats de données pris en charge, nous retrouvons les formats classiques des plateformes de formation, c'est-à-dire le stockage en interne dans les bases de données (SQL) et les formats d'import/export avec notamment le CSV. Ce bilan démontre qu'il reste un travail conséquent à mener sur l'interopérabilité et l'utilisation d'environnements standardisés comme xAPI, le partage de résultats et leur réutilisabilité ainsi que la prise en compte des autres usagers comme les apprenants.

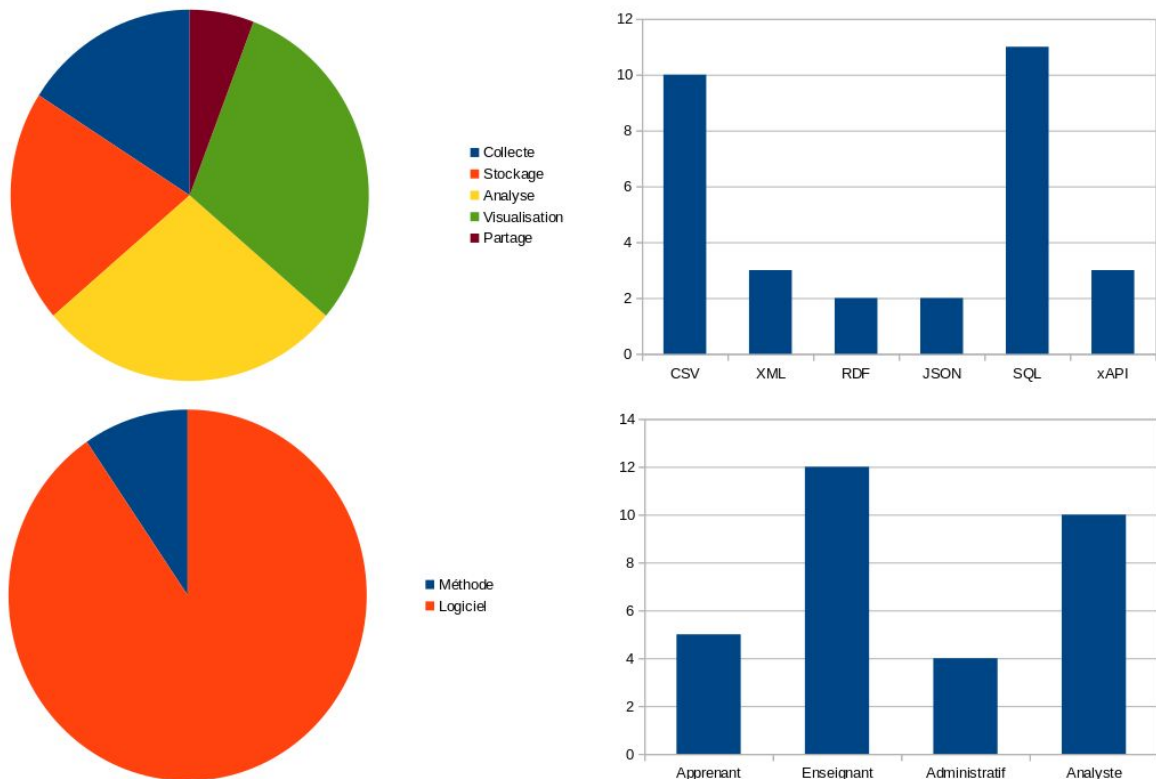


Figure 14. Analyse des 21 outils retenus issus de la Recherche Française

4 - Présentation des principaux outils

DDART

DDART (Dynamic Dashboard for collection, Analysis and visualization of activity and Reporting Traces) [Michel, et al, 2017] est un outil qui peut s'intégrer à un ENT conçu avec Moodle. L'intégration de DDART offre, en complément des outils de gestion du travail collaboratif (forum, chat, wiki, ...) de Moodle, deux outils de planification et de suivi de l'activité des élèves et du groupe :

- le *reporting tool* permet de spécifier différents objectifs à atteindre, de décrire comment l'activité se réalise ou de juger la qualité de la réalisation intrinsèque ;
- le *dashboard* offre une vue globale sur des indicateurs de suivi des activités rapportées et des activités réalisées avec les outils de l'ENT. Il permet ainsi de suivre qui travaille sur quoi, combien de temps, quel est le jugement des uns et des autres... Ces indicateurs sont créés dynamiquement par les élèves ou l'enseignant en fonction des besoins du projet pédagogique.

Le *reporting tool* permet aux élèves de collecter des éléments d'information sur leur activité, de gérer leurs rapports d'activité, de les modifier *a posteriori* ou faire des commentaires généraux sur les contextes des activités. Les élèves peuvent également lire et commenter les rapports des autres élèves du groupe, ce qui est très utile pour que l'élève se situe par rapport au groupe et puisse aussi prendre du recul sur son activité.

Figure 15. DDART - Saisie d'un rapport d'activité

La rédaction d'un nouveau rapport se fait sur la base de séquences de phrases semi-structurées, proposées dans une base de modèles créés par l'enseignant. Actuellement trois modèles existent : décrire ce que l'on fait, décrire ce que l'on pense, auto-évaluer la réalisation d'un objectif (la qualité d'une réalisation/production ou l'atteinte d'une compétence). La saisie se fait en complétant les phrases à trous (voir Figure 14). Cette semi-structure permet de produire des rapports lisibles et de collecter des items qualifiés sur l'activité. Une interface de gestion des rapports permet de les éditer (voir Figure 15) ou de les lire (voir Figure 16).

Write a new report **Manage my reports** Read other reports

Title	Date	Privacy	Edit
my goal	12-1-2014 23:12:44	public	
12/1 activity report	12-1-2014 23:42:42	public	
13-1 report	13-1-2014 23:24:27	public	
14-1 report	14-1-2014 23:42:20	public	

2 pages(1/2) [1][2]

Report content

I schedule to read an e-book from 12-01-2014 09:00 to 12-01-2014 09:30 .

I schedule to discuss from 12-01-2014 16:00 to 12-01-2014 16:20 .

My php should reach the level of good .

My mysql should reach the level of very good .

My documentation should reach the level of neutral .

My communication should reach the level of good .

Figure 16. DDART - Gestion de ses propres rapports

Write a new report

Manage my reports

Read other reports

Title	Date	Writer	Detail
goal report	12-1-2014 00:17:40	Julien	
11/1 report	12-1-2014 00:29:28	Julien	
12/1 report	12-1-2014 17:34:27	Julien	
my goal report	12-1-2014 22:33:46	Benoit	
first activity report	12-1-2014 22:55:45	Benoit	

5 pages(1/5) [1][2][3][4][5]

Report content

I organize a meeting with Benoit,Sophie,Francois,Pascal about distribution of work using Skype from 11-01-2014 09:00 to 11-01-2014 09:30 at home . I judge that Benoit is good because he is active in the meeting . I self assess that my mood is very good because this project is interesting .

Figure 17. DDART - Lecture des rapports des autres

Le tableau de bord (*dashboard*) offre une possibilité de création d'indicateurs sur la base des traces de l'activité collectées manuellement à partir du *reporting tool* et des traces de l'activité enregistrées automatiquement s'il y a usage de Moodle dans la formation. L'interface de conception des indicateurs permet, par des *drags and drops* entre les zones "paramètre" et "design d'indicateur", de spécifier : les entités (les éléments sur lesquels se font les calculs) ; les types de données (fréquence, intervalle de temps, contenu, description) liées aux entités ; les types de calculs ; et enfin les types de visualisations. La spécification de tout nouvel élément dans la zone "indicator design" provoque un affichage direct du visuel de l'indicateur dans la zone centrale du tableau de bord. Tous les éléments sont modifiables à tous moments. Ce calcul dynamique permet à l'utilisateur d'adapter facilement la conception de l'indicateur pour atteindre la forme voulue.

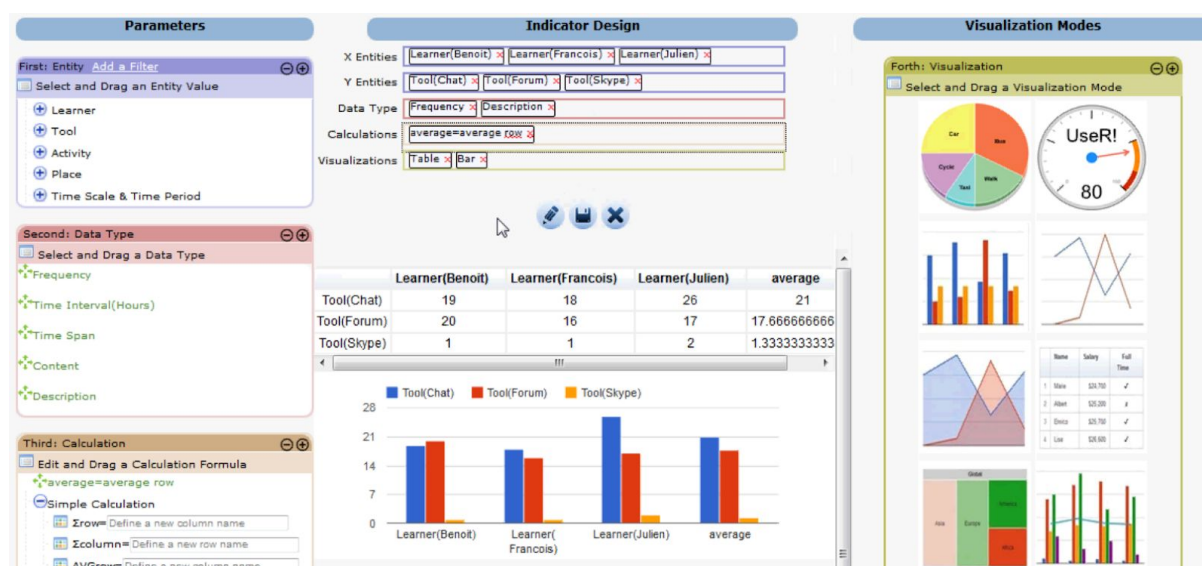


Figure 18. DDART - Interface de création du tableau de bord

L'indicateur peut être enregistré en mode privé (et ainsi être uniquement accessible à l'utilisateur) ou en mode public (et être accessible au groupe). Le tableau de bord offre une interface de visualisation de l'ensemble des indicateurs créés.

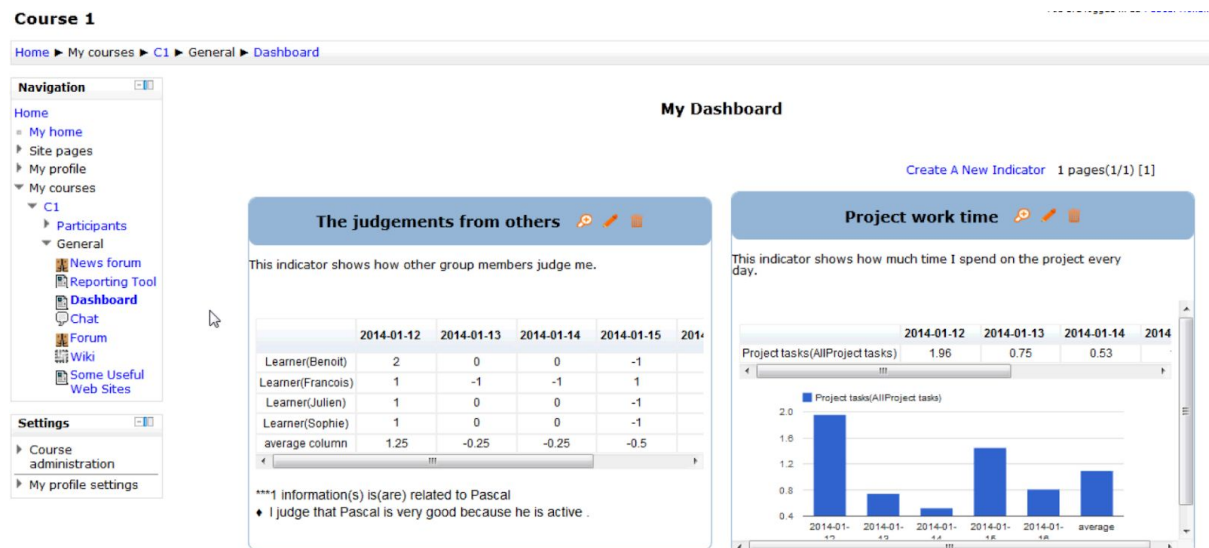


Figure 19. DDART - Exemple de tableau de bord

Laalys

1. Principe de fonctionnement de Laalys

Laalys [Muratet et al., 2016][Yessad et al., 2017] est un système de suivi d'apprenants dans des jeux sérieux (et plus généralement dans des environnements d'apprentissage interactifs). Il implémente un algorithme qui exploite le formalisme des réseaux de Petri (RdP) afin d'associer des étiquettes pédagogiques aux actions de l'apprenant et calculer un score à partir de ces étiquettes. Ces dernières renseignent les utilisateurs et notamment les enseignants sur le comportement d'un apprenant en qualifiant par exemple ses actions comme étant correctes, erronées, tardives ou prématurées.

Un réseau de Petri décrivant la ou les procédures de résolution préconisées par les experts est construit. Ce réseau de Petri, une fois validé par les experts, est utilisé pour analyser l'écart entre la solution de l'apprenant et celles des experts. Le principe de l'algorithme d'étiquetage est de fournir des informations sémantiques sur les écarts détectés entre la résolution de l'apprenant et celles des experts.

La figure 20 illustre les différents processus permettant de mettre en place le suivi des apprenants. D'une part, nous avons l'algorithme de Laalys qui permet d'étiqueter le processus de résolution d'un apprenant. D'autre part, nous avons le système auteur qui assiste les enseignants/ concepteurs de jeux sérieux dans la construction du réseau de Petri modélisant la ou les résolutions préconisées par les experts (noté RdPFiltré sur le schéma de la figure 20).

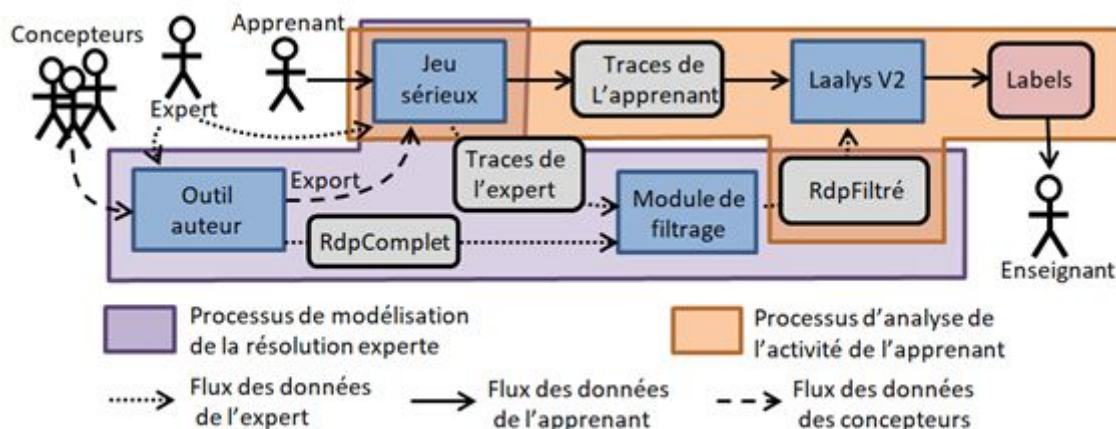


Figure 20. Architecture générale des processus de traitement de Laalys

2. Algorithme d'étiquetage de Laalys

Les entrées de l'algorithme d'étiquetage sont le RdpFiltré et la trace d'un apprenant. L'algorithme d'étiquetage s'appuie sur l'identification des 3 cas suivants : le premier cas correspond aux situations dans lesquelles l'apprenant a réalisé des actions qui sont refusées par le jeu et donc qui ne modifient pas l'état du jeu ; le deuxième cas correspond aux situations où l'apprenant a réalisé des actions qui sont acceptées par le jeu et donc en modifient l'état ; le troisième et dernier cas correspond à l'identification des actions manquantes alors que le jeu est terminé.

Chacun de ces cas mène à une description qui permet de caractériser l'action de l'apprenant au moyen d'une (et d'une seule) étiquette pédagogique.

2.1. Cas 1 - actions qui ne changent pas l'état du jeu

Ce cas apparaît lorsque l'apprenant essaie de réaliser une action qui n'est pas autorisée dans l'état actuel du jeu. Par exemple, si le joueur tente d'ouvrir une porte alors qu'il n'a pas la bonne clé. Nous avons identifié quatre étiquettes spécifiques pour caractériser ces actions de manière plus précise : (1) l'action était disponible dans le graphe d'accessibilité du RdpFiltré, l'étiquette générée pour étiqueter cette action est « *Tardive* », (2) l'action sera disponible dans le graphe d'accessibilité du RdpFiltré, l'étiquette générée est « *Prématurée* », (3) l'action a été disponible et le sera à nouveau plus tard dans le RdpFiltré, l'étiquette générée dans ce cas est « *Non synchronisée* », ou enfin (4) cette action n'est jamais disponible dans le graphe d'accessibilité du RdpFiltré, i.e. elle n'est jamais réalisée par les experts dans la résolution du niveau, l'étiquette générée dans ce quatrième cas est « *Erronée* ».

2.2. Cas 2 - actions qui changent l'état du jeu

Cela correspond au cas le plus courant, mais toutefois toutes les actions ne sont pas forcément correctes en comparaison avec la résolution des experts. En effet, la plupart des jeux pour apprendre permettent à l'apprenant de faire des erreurs. Ce principe d'essai/erreur est au cœur des jeux vidéo [Gee, 2007]. L'opportunité de réaliser des séquences d'actions inutiles, redondantes ou incorrectes dans un jeu sérieux [Young, 2009] [Thomas et al., 2011] est fondamentale afin d'initier un processus de réflexion libre chez les apprenants et tenter d'atteindre les objectifs pédagogiques visés. L'algorithme évalue donc si le nouvel état courant du jeu quitte l'espace d'état filtré (que nous appellerons f-space et représente le

graphe d'accessibilité du RdpFiltré), revient dans le f-space, se déplace à l'intérieur du f-space ou se déplace en dehors du f-space.

- Cas 2.1 - sortie du f-space

Ce cas peut apparaître quand l'apprenant joue une action qui n'a pas été réalisée par les experts. Si l'état du jeu est en-dehors du f-space, cela signifie que l'état du jeu généré ne correspond à aucun des états présents dans le f-space. L'étiquette utilisée pour étiqueter cette action est « *Erronée* ».

- Cas 2.2 - retour vers le f-space

Cela correspond au moment où l'apprenant corrige de précédentes erreurs et retourne dans le f-space. Dans ce cas, l'algorithme (1) vérifie si l'état du jeu résultant de l'action de l'apprenant est inclus dans l'historique des états déjà atteints par l'apprenant, (2) calcule la longueur du plus court chemin entre ce nouvel état et un état final et (3) évalue la distance minimale (plus court chemin) entre chaque état de l'historique et un état final. En fonction des résultats sur ces trois critères, les étiquettes suivantes sont générées pour l'action de l'apprenant : « *Rattrapage* » ($1 \wedge (2=3)$), « *Retour arrière* » ($1 \wedge (2>3)$), « *Progression* » ($\neg 1 \wedge (2<3)$) ou « *Non optimale* » ($\neg 1 \wedge (2\geq 3)$).

- Cas 2.3 - déplacement dans le f-space

Dans ce **cas**, l'algorithme vérifie si l'apprenant progresse vers un état final ou tend à s'en éloigner. Ainsi l'algorithme (1) calcule la longueur du plus petit chemin entre le nouvel état et un état final, (2) évalue la longueur du plus court chemin entre l'état précédent et un état final, (3) vérifie si ces deux chemins sont différents, (4) vérifie si l'action effectuée est une action experte et (5) vérifie si le nouvel état est un successeur direct de l'état précédent dans le f-space. En fonction des résultats sur ces cinq critères, les étiquettes suivantes sont générées pour l'action de l'apprenant : « *Correcte* » ($((1<2)\wedge 4)$), « *Equivalente* » ($((1<2)\wedge \neg 4\wedge 5)$), « *Progression* » ($((1<2)\wedge \neg 4\wedge \neg 5)$), « *Inutile* » ($((1=2)\wedge \neg 3)$), « *Non-optimale* » ($((1=2)\wedge 3)$), « *Régression* » ($1>2$).

- Cas 2.4 - déplacement en dehors du f-space

Ce comportement correspond au cas où l'apprenant réalise une action qui conserve le jeu dans un état en-dehors du f-space. Dans ce cas, l'algorithme essaie de calculer un nouveau f-space en initialisant le RdpFiltré à l'état courant du jeu. Si un état final est atteignable à partir de l'état courant, l'algorithme traite cette action comme dans le cas 2.3. Sinon, l'algorithme détermine si le nouvel état est plus proche (l'étiquette générée est « *Rapprochement* »), plus loin (l'étiquette générée est « *Eloignement* ») ou équidistant (l'étiquette générée est « *Stagnation* ») d'un état final du niveau que l'état qui le précède.

2.3. Cas 3 - actions manquantes lorsque le jeu est terminé

Si la dernière action de l'apprenant ne lui permet pas d'atteindre un état final du jeu, l'algorithme recherche le dernier état dans le f-space atteint par l'apprenant et calcule le chemin le plus court pour atteindre la fin du niveau à partir de cet état. Ensuite, l'algorithme étiquette les actions de ce chemin comme « *Manquantes* ».

3. Calcul du Score d'un apprenant dans Laalys

L'algorithme d'étiquetage, présenté ci-dessus, produit 16 étiquettes pédagogiques. Nous associons un coefficient pour chaque étiquette afin d'être en mesure de proposer un score pour l'apprenant (voir Tableau 1). Tout d'abord, nous définissons comme des étiquettes positives, celles qui garantissent une progression vers un état final. Nous définissons les coefficients associés à ces étiquettes dans l'intervalle $]0 ; 1]$. Pour cette classe, nous retenons les étiquettes suivantes : « *Correcte* » dénote un comportement satisfaisant par

rapport à la solution des experts (coeff=1) ; « Equivalente » est similaire à une action « Correcte », mais fait référence à une action non-experte (d'où un coefficient plus bas : 0,8) ; « Progression » signifie que l'apprenant suit un processus de résolution non-expert mais se rapproche de la solution (coeff=0,6) ; et « Rapprochement » est associée au plus petit coefficient positif car cela arrive lorsque l'apprenant se déplace en dehors du f-space (coeff=0,4).

Tableau 1 • Coefficients des étiquettes positives et négatives

Étiquettes positives	Coeff.	Étiquettes négatives	Coeff.
Correcte	1	Inutile	-0,1
Équivalente	0,8	Eloignement	-0,2
Progression	0,6	Manquante	-0,3
Rapprochement	0,4	Erronée	-0,5

Ensuite, nous définissons comme étiquettes négatives, celles qui représentent un écart sûr par rapport à la solution des experts. Afin de donner plus de poids aux étiquettes positives dans le calcul du score et moins pénaliser les essais-erreurs, nous définissons les coefficients négatifs dans l'intervalle $[-0,5 ; 0]$. Pour cette classe, nous retenons les étiquettes suivantes : « Erronée » est l'opposée de « Correcte » (coeff=-0,5) ; « Manquante » apparaît quand les apprenants abandonnent le niveau, nous choisissons de pondérer fortement cette étiquette pour dégrader le score des étudiants dans le cas d'un abandon (coeff=-0,3) ; « Éloignement » est l'opposée de « Rapprochement » (coeff=-0,2) ; et « Inutile » ne caractérise pas une erreur majeure et a donc une influence mineure sur le score (coeff=-0,1).

Finalement, les étiquettes restantes (Non-optimale, Rétablissement, Retour arrière, Stagnation, Prématurées, Tardives, Régression et Non synchronisée) sont classées comme étiquettes neutres. Même si leurs coefficients sont mis à 0, elles vont contribuer au calcul du score en augmentant la taille de la trace et dégradent ainsi le score de l'apprenant.

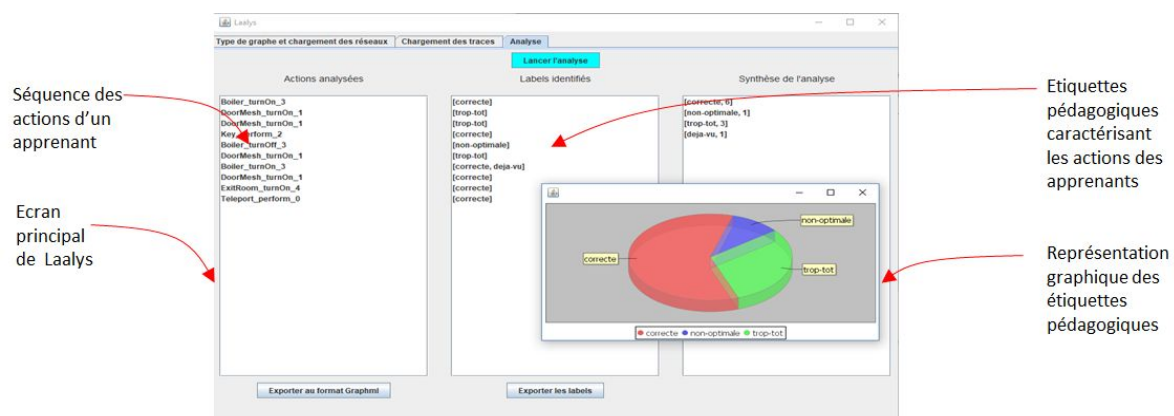


Figure 21. Capture d'écran de l'IHM de Laalys

UTL - Usage Tracking Language

UTL (Usage Tracking Language, [Iksal, 2012; Iksal, 2011]) est un environnement global d'analyse de traces pédagogiques basé sur la prescription. Il repose sur un cycle complet (cf. Figure 22) qui nécessite d'identifier au préalable les besoins d'observation exprimés par les acteurs de la situation d'apprentissage, modélise les indicateurs, réalise le calcul et retourne le résultat sous la forme d'un tableau de bord.

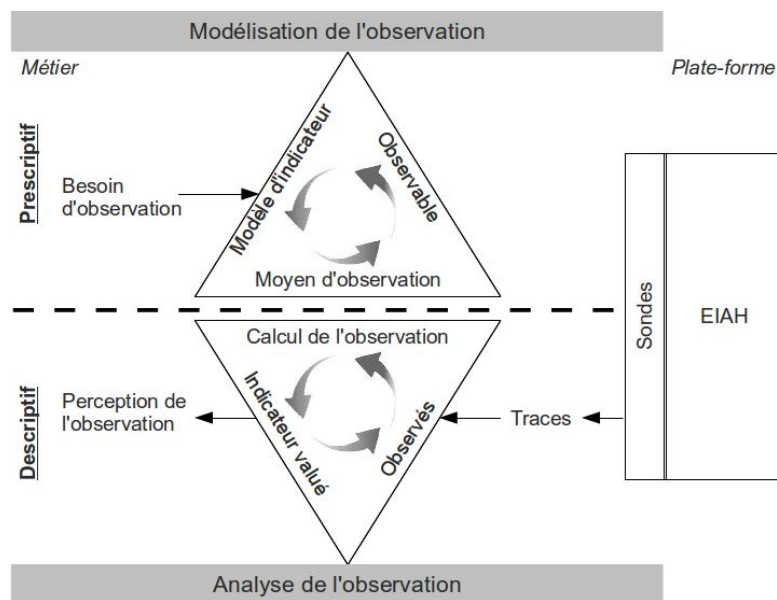


Figure 22. Cycle d'observation selon UTL

UTL a été conçu selon deux principes, tout d'abord, le processus d'observation est élaboré pour et avec les usagers, c'est pour cela que l'étape d'identification des besoins par les usagers est primordiale. Ensuite, le processus d'analyse en tant que tel est indépendant d'une plateforme d'apprentissage ainsi que d'un format de base de données spécifique. En effet, UTL embarque son propre langage de calcul dont les formules reposent sur un format de représentation interne de l'information. Ce choix technique facilite la réutilisation des processus au sens où "il suffit" que les données au sens sémantique du terme soient présentes dans la plateforme d'apprentissage ou plutôt dans le processus de collecte des traces. Un système d'importation permet de transformer des traces exprimées aux formats XML, SQL, JSON, CSV ou simplement textuels, dans le format interne d'UTL.

La définition d'un indicateur pédagogique au sens de notre plateforme est qu'il s'agit d'une information qui fait sens à l'utilisateur dans son contexte, cette information est calculée à partir des traces et/ou données issues de la plateforme informatisée d'apprentissage. Pour cela, la gestion des données au sens d'UTL repose sur la séparation entre données primaires et données dérivées (cf. Figure 23). Les données primaires sont bien entendu les traces brutes fournies par le système (base de données, fichiers de logs, ...) mais aussi des données additionnelles (préexistantes le plus souvent à la situation d'apprentissage comme un scénario pédagogique, un emploi du temps, une liste d'apprenants ...) et des données de production d'utilisateurs (productions directement exploitables ou transformées pour être utilisées numériquement comme des évaluations de travaux). Les données dérivées sont

des données qui résultent d'une transformation, d'un calcul depuis des données primaires et/ou des données dérivées. Elles se décomposent entre données intermédiaires (pouvant servir au calcul de différentes données) et les indicateurs à destination des usagers.

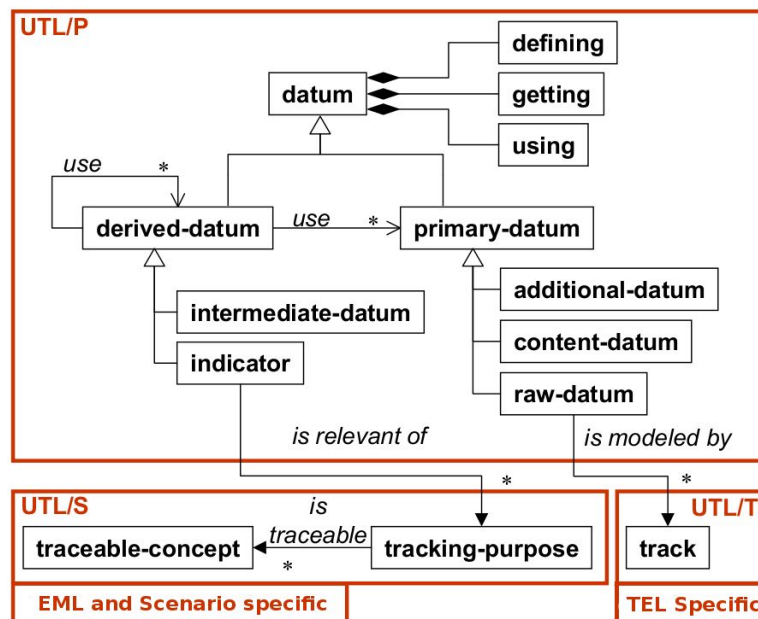


Figure 23. Les données d'UTL

Le calculateur des données dérivées exploite son propre langage de calcul (cf. Figure 24) [PhamThi Ngoc, 2011] afin de faciliter la capitalisation et la réutilisation des processus de calcul. De plus, à des fins d'interopérabilité, le calculateur autorise l'exécution de programmes externes pour les algorithmes plus efficaces avec d'autres outils.

```
<DCL4UTL-formula>
cal {
  for each $c in ID3.chapter as sort(ID3.chapter[startAt] asc) do {
    $name=externe Externals extraireAttribut(c,"name");
    for each $l in ID2.User do {
      $usr = externe Externals extraireContenu(l);
      $cntv = ID1.countVideo as filter(ID1.countVideo[learner]==usr and ID1.countVideo[chapter]==name);
      $cptv = externe Externals extraireContenu(cntv);
      $cntq = ID4.countQuizz as filter(ID4.countQuizz[learner]==usr and ID4.countQuizz[chapter]==name);
      $cptq = externe Externals extraireContenu(cntq);
      if ((cptv > 0) and (cptq > 0)) {
        IDR.activeViewer[learner=usr][chapter=name]=usr;
      }
    }
  }
}
where
  ID1 = ID_CountVideo.using.data.intermediateDatum,
  ID2 = ID_Learners.using.data.intermediateDatum,
  ID3 = ID_Chapters.using.data.intermediateDatum,
  ID4 = ID_CountQuizz.using.data.intermediateDatum,
  IDR = ID_ListActiveViewers.using.data.intermediateDatum
</DCL4UTL-formula>
```

Figure 24. Exemple de formule de calcul sous UTL

UTL Editeur v2.0

Accueil Projets Déconnexion

ID_LogStartStop.xml

Définition (Defining) Obtention (Getting) Utilisation (Using)

Title *

Liste des logs des utilisateurs concernant le début et la fin des modules

Cardinality *

1 or n

Description

Save Back

Figure 25. Copie d'écran de l'éditeur UTL

La plateforme UTL dispose d'un éditeur pour les données (cf. Figure 25), d'un outil d'importation de différents formats de traces en données brutes (Texte, SQL, CSV, JSON, XML) et d'un calculateur. Le résultat des indicateurs peut être exploité dans un tableau de bord construit adhoc [Loup, 2016] (cf. Figure 26) ou avec un générateur de tableau de bord [Dabbebi, 2017]. Actuellement, les résultats sont stockés dans une base de données noSQL (eXist-db) qui permet l'interrogation depuis de nombreux langages de programmation.

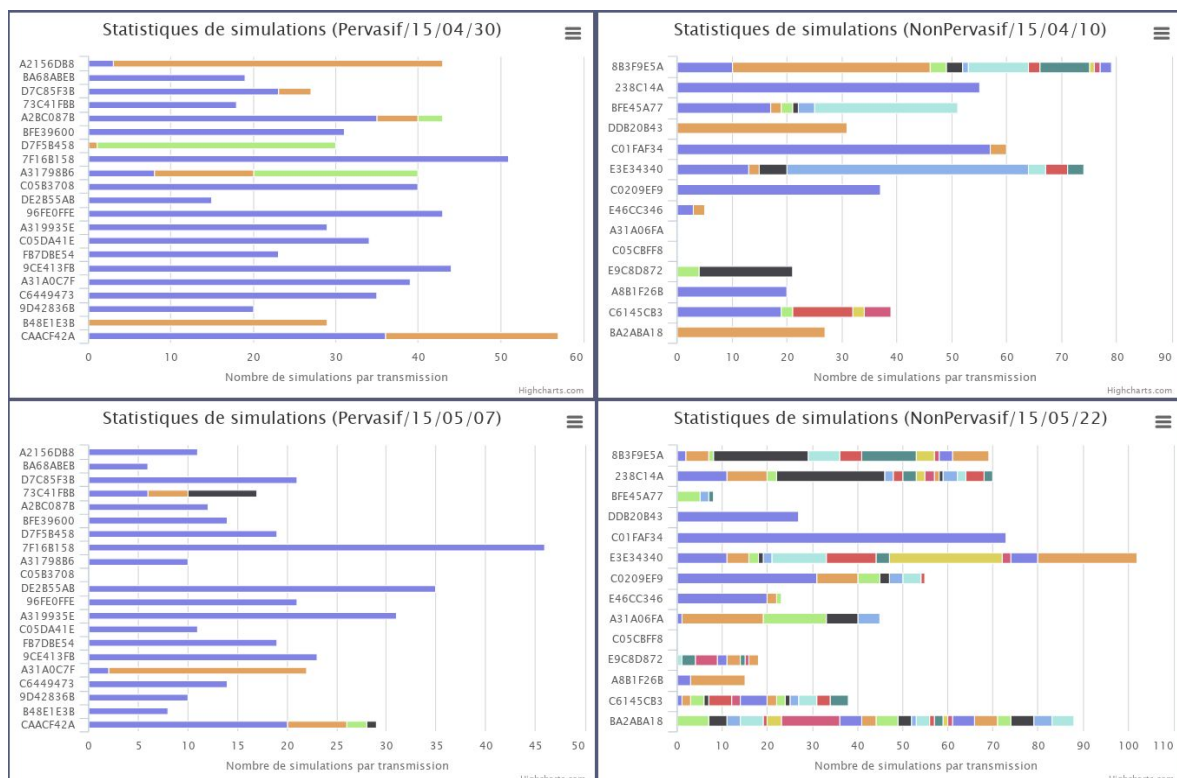


Figure 26. Exemple d'affichage d'indicateur UTL

Lab4CE

Les laboratoires virtuels et distants (Virtual and Remote Laboratories, VRL) sont des environnements d'apprentissage dédiés à l'apprentissage exploratoire, appliqué dans des activités pratiques médiatisées par l'Informatique. Dans les EIAH (Environnements Informatiques pour l'Apprentissage Humain) pour les VRL, les apprenants développent des compétences liées à la démarche scientifique à travers leurs interactions avec les dispositifs distants physiques ou simulés, mais également des compétences de travail en équipe par les interactions qu'ils ont entre eux ou avec les enseignants. Par le suivi de ces interactions, ces EIAH permettent d'explorer les comportements des apprenants à un niveau de détail élevé, et peuvent donc offrir une meilleure compréhension du processus d'apprentissage qu'ils mettent en œuvre.

Vue générale de l'environnement

Lab4CE est une plateforme web s'appuyant sur un gestionnaire de cloud pour déployer à la demande des laboratoires informatiques virtuels, et munie d'un ensemble de fonctionnalités support à l'apprentissage [Broisin et al., 2017a]. Ce système a été conçu notamment pour s'affranchir des limites spatiales et temporelles, et des restrictions d'accès aux ressources informatiques. La plateforme permet de fournir à chaque apprenant un ensemble d'équipements virtuels accessibles en tout lieu, à tout moment, sans limite d'utilisation (i.e., les étudiants sont administrateurs de leurs équipements).

Dans notre environnement, les instructeurs peuvent concevoir un TP en définissant une topologie composée de machines et d'équipements d'interconnexion. Lorsqu'un apprenant se connecte pour une activité particulière, et accède la première fois à son laboratoire, le système crée et configure automatiquement l'ensemble des ressources propres à l'étudiant. Celui-ci peut alors manipuler les machines (i.e., les démarrer, les mettre en veille ou les arrêter) et interagir avec celles-ci via un terminal embarqué dans l'interface web, similaire à un terminal traditionnel. L'environnement propose des fonctionnalités dédiées à l'apprentissage en s'appuyant sur un framework de collecte et d'analyse de données qui permet, à partir des interactions de l'apprenant avec le système, de forger différentes traces d'apprentissage et de les enrichir à la volée par le calcul d'indicateurs définis par l'enseignant ou le concepteur pédagogique [Broisin et al., 2017b].

Framework et workflow d'analyse de données

La plupart des interactions de l'utilisateur avec la plateforme sont collectées : les connexions au système, les messages instantanés, les demandes d'aide, les actions sur les laboratoires et les ressources, et la navigation entre les différentes interfaces de la plateforme. Concernant les interactions entre les utilisateurs et les terminaux des machines, chaque entrée dans le terminal est tracée, qu'elle provienne de l'utilisateur lorsqu'il écrit sur son clavier ou de la machine lorsque celle-ci répond.

Le framework illustré par la Figure 27 a pour objectif la génération et le stockage d'enregistrements xAPI, ainsi que leur enrichissement par l'inférence d'indicateurs [Venant et al., 2016]. Il s'inspire d'infrastructures existantes telles que celles proposées dans le projet Migen [Gutierrez-Santos et al., 2010] ou dans l'approche flexible et extensible de [Hecking et al., 2014]. Cependant, à la différence de ces approches, notre architecture réside

essentiellement côté client (i.e., dans le navigateur web de l'utilisateur) afin de bénéficier du potentiel de calcul distribué sur les postes clients connectés à la plateforme.

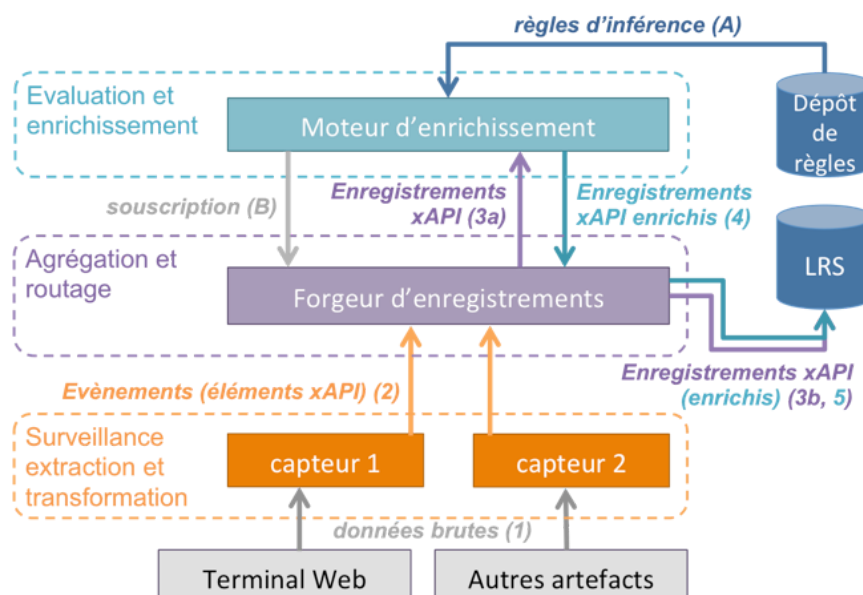


Figure 27. Framework de traces

Notre proposition inclut trois couches de composants faiblement couplés côté client, ainsi que deux dépôts côté serveur. Les capteurs surveillent les flux de données sur des composants spécifiques du système ((1) sur la Figure 27), génèrent des éléments xAPI et les envoient au forgeur d'enregistrements sous forme d'événements (2). Le forgeur agrège les différents éléments en enregistrements xAPI, qui sont alors routés soit au moteur d'enrichissement (3a), soit directement au dépôt de traces (3b). Après une phase initiale où le moteur d'enrichissement souscrit au forgeur (B) pour recevoir les enregistrements qu'il peut enrichir d'après les règles décrites dans le dépôt de règles (A), il infère et ajoute à chaque enregistrement un ou plusieurs indicateurs, et les renvoie au forgeur (4) avant leur stockage dans le LRS (5).

Les interactions entre les apprenants et leurs machines sont essentiellement des instructions Shell exécutées via le terminal ; elles sont constituées d'une commande et, le cas échéant, d'options et d'arguments (ex. : `ls -a -l` est l'exécution de la commande `ls` avec les arguments `-a` et `-l`). À l'issue de son exécution, la machine peut retourner une réponse textuelle (ex. : pour la commande `ls -a -l`, la réponse correspond à la liste détaillée de tous les fichiers et répertoires du répertoire courant).

Ainsi, lorsqu'un étudiant interagit avec un terminal d'une machine, le framework capture chaque caractère qui apparaît au sein du terminal (i.e., les touches tapées par l'apprenant, et les caractères retournés par la machine). Le moteur est capable de reconstituer les commandes exécutées, en distinguant la commande de ses arguments et de sa réponse. Grâce au moteur d'inférence, un ensemble de règles permettent de qualifier la justesse technique d'une commande à partir des informations précédentes, c'est-à-dire de calculer une valeur booléenne indiquant si la commande a été exécutée sans erreur par la machine [Venant et al., 2016]. Un enregistrement xAPI d'une commande comprend donc : (i) l'horodatage, (ii) l'identité de l'apprenant, (iii) l'identifiant du laboratoire, (iv) celui de la machine, (v) le nom de la commande, (vi) ses arguments, (vii) la réponse de la machine suite à l'exécution de la commande, et (viii) la justesse technique de la commande.

A partir de ces données, différents outils de visualisation peuvent ainsi être conçus : un outil de comparaison sociale, un outil de réflexion a posteriori des activités, ainsi qu'une visualisation exposant à l'apprenant les stratégies d'apprentissage qu'il met en œuvre.

L'outil de comparaison sociale

L'outil de visualisation que nous avons conçu affiche un ensemble de barres de progression qui reflète le niveau de performance des apprenants (d'après la justesse technique d'une instruction) par un simple code couleur (vert si la valeur est vraie, rouge sinon). Une barre de progression est un composant léger qui souscrit au LRS pour recevoir un flux d'enregistrements xAPI d'instructions à propos d'un apprenant ou d'un groupe d'apprenants. L'outil dessine alors un dégradé de couleurs au fil des enregistrements reçus, à la manière d'un spectrogramme, graphiquement réparti dans le temps.

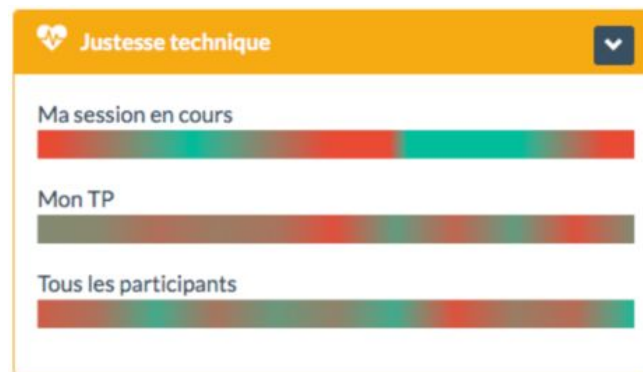


Figure 28. L'outil de comparaison sociale

Cet outil illustré par la Figure 28 est intégré dans Lab4CE et fournit trois barres de progression distinctes. La première (Ma session en cours) reflète la performance individuelle de l'apprenant connecté à la plate-forme pendant sa session courante ; la seconde (Mon TP) exprime cette performance depuis la première action de l'apprenant dans le TP ; la dernière (Tous les participants) expose le niveau de performance moyen du groupe d'apprenants inscrit à ce TP. Les barres sont automatiquement mises à jour à chaque nouvelle commande exécutée par un apprenant. Cet outil de comparaison sociale permet aux apprenants de prendre conscience de leur propre performance, mais aussi de celle de leurs pairs, tout en nécessitant un faible effort cognitif leur permettant de rester concentrés sur leur activité. De plus, les tuteurs sont conscients du niveau de performance du groupe et sont donc plus à même d'ajuster les objectifs et/ou le déroulé du TP.

Réflexion a posteriori de l'activité d'apprentissage

Afin de permettre aux apprenants d'analyser en détail leurs propres actions ainsi que celles de leurs pairs, nous proposons une interface illustrée par la Figure 29 qui permet aux apprenants d'affiner l'analyse de leur propre travail ou celui de leurs pairs. Le formulaire en haut de l'interface offre un ensemble de filtres pour visualiser une session selon un utilisateur, une date ou une ressource. Pour chaque ressource sélectionnée, une chronologie des instructions exécutées est représentée. Chaque nœud de la chronologie représente une instruction ; celui-ci est coloré selon sa justesse technique, tandis que le détail de l'instruction apparaît dans un simili-terminal lorsque le curseur est positionné sur le nœud correspondant.

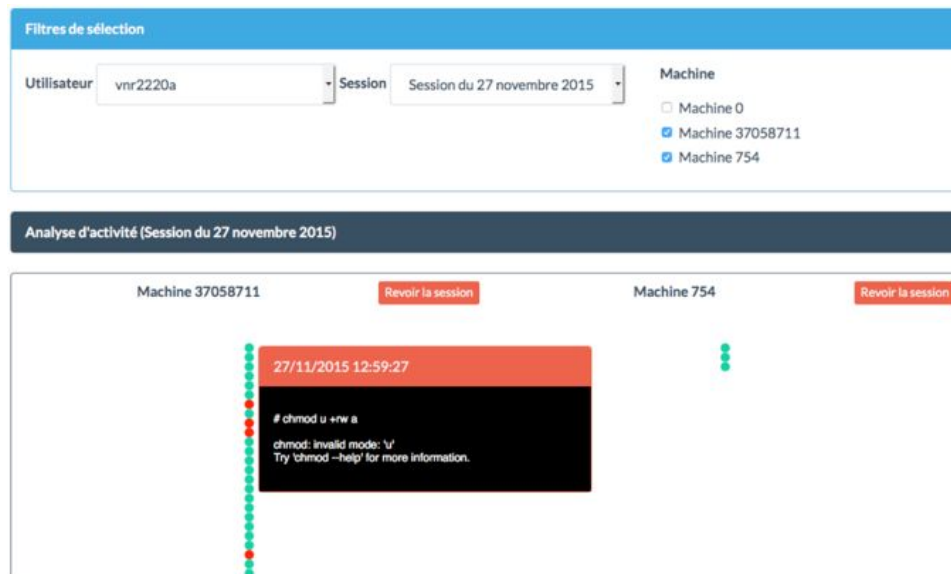


Figure 29. L'outil de réflexion a posteriori

Analyse des stratégies mises en œuvre

Si l'étude des actions des apprenants à travers différentes mesures d'activité est une première approche, l'analyse de motifs séquentiels peut fournir une autre compréhension de leurs comportements [Aleven et al., 2006]. Nous nous sommes donc appuyés sur les méthodes d'analyse des usages pour identifier de nouveaux facteurs de succès de l'apprentissage. Son originalité réside dans l'exploration des liens existants entre le comportement des apprenants pendant des activités de travaux pratiques et leur performance académique. À partir de traces issues d'expérimentations menées en contexte d'apprentissage réel, nous avons découvert un ensemble de motifs d'actions récurrents et défini des stratégies d'apprentissage de plus haut niveau d'abstraction [Venant et al., 2017a]. Les résultats montrent l'existence de corrélations entre certaines stratégies et la performance des apprenants : la construction progressive d'une action complexe, ou la réflexion avant l'exécution d'une action, sont deux stratégies appliquées plus fréquemment par les étudiants performants au test final. La Figure 30 permet aux apprenants de visualiser l'évolution des stratégies qu'ils mettent en œuvre tout au long de la réalisation des activités d'apprentissage.

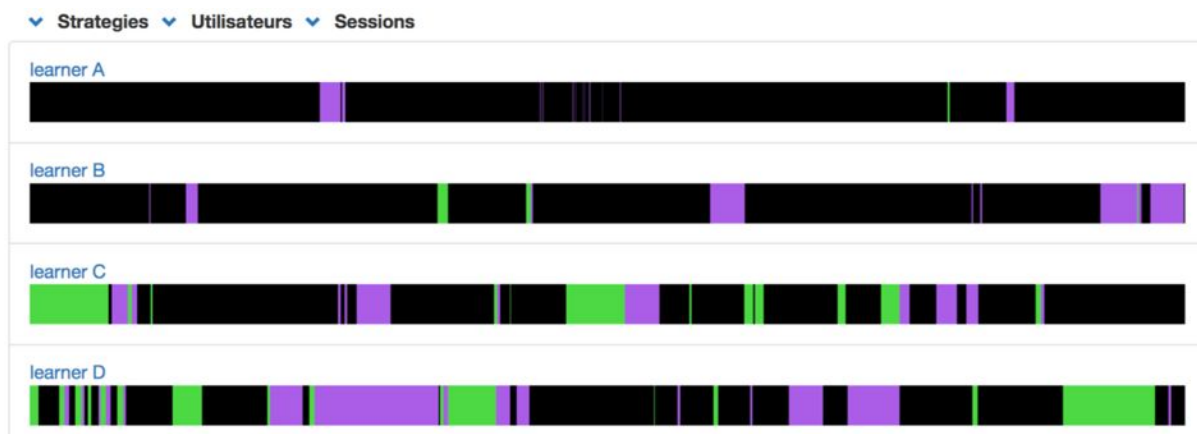


Figure 30. Suivi du comportement des apprenants en temps réel

À partir de ces résultats, nous avons intégré à la plateforme Lab4CE deux nouvelles fonctionnalités fondées sur deux patrons de conception différents [Venant et al., 2017b]. La première s'appuie sur un tuteur intelligent capable de guider les apprenants pendant leurs travaux pratiques selon les stratégies d'apprentissage qu'ils adoptent. Par exemple, lorsqu'un étudiant applique la stratégie tâtonnement sur une commande donnée, le tuteur lui conseille de consulter le manuel de cette commande dans le but de placer l'apprenant dans des stratégies liées à la réflexion. Aussi, lorsque des apprenants ont déjà utilisé le manuel mais continuent d'échouer à l'utilisation de la commande, le système leur suggère de chercher de l'aide auprès d'un pair qui a déjà exécuté la commande avec succès. La seconde fonctionnalité est un système d'awareness signalant aux enseignants, à partir de l'analyse des stratégies mises en œuvre par les apprenants, ceux qui semblent être en situation de faiblesse. Par exemple, lorsque différents apprenants tâtonnent sur la même commande, le système prévient l'enseignant de la difficulté rencontrée sur la commande concernée afin de favoriser une intervention collective.

KTBS4LA

La plateforme d'analyse de traces KTBS4LA [Casado et al., 2017] (kernel for Trace-Based Systems for Learning Analytics) a pour objectif de faciliter l'utilisation des systèmes à bases de traces proposés par l'équipe TWEAK du laboratoire LIRIS. Ces systèmes à base de traces offrent plusieurs outils et méthodes pour transformer et analyser les traces d'interaction.

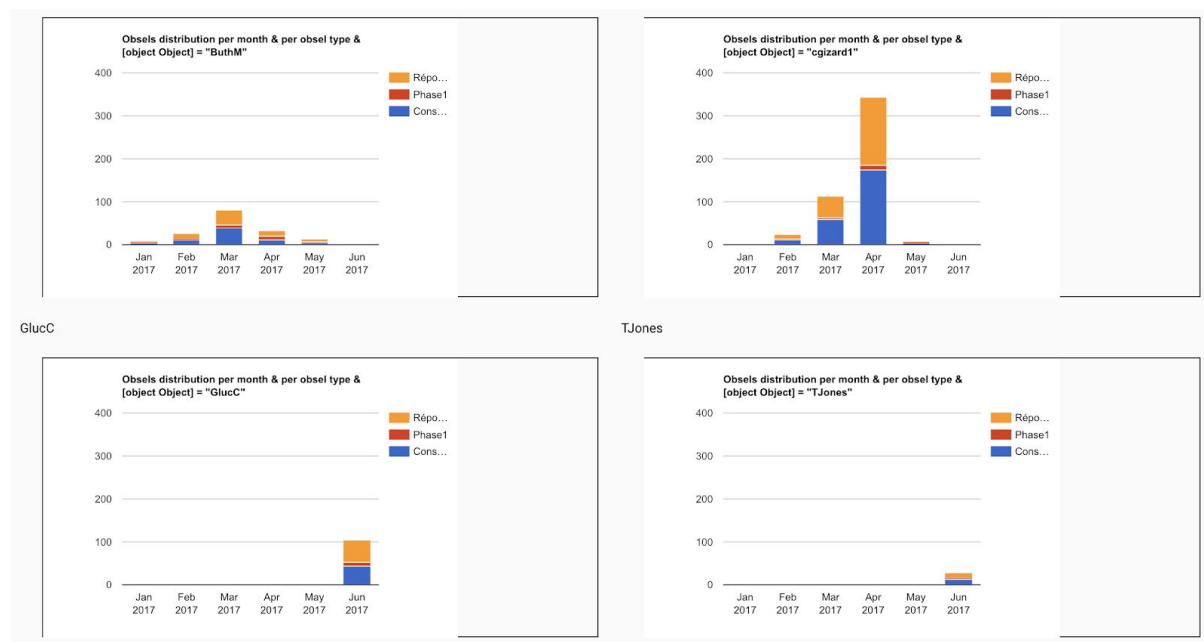


Figure 31. KTBS4LA - Visualisation de traces sous forme d'histogramme

KTBS4LA se présente sous la forme d'une application web et permet une manipulation graphique des données collectées. La plateforme repose sur une modélisation des traces, c'est-à-dire que chaque élément collecté lors de l'activité de l'élève est associé à un modèle

décrivant le type de l'action effectuée et les attributs qui la caractérisent. Grâce à cette modélisation, KTBS4LA propose plusieurs outils de visualisation.

Une première visualisation consiste à utiliser des histogrammes pour avoir un aperçu du contenu des traces. L'utilisateur peut choisir selon quelle plage de temps il souhaite découper l'information et choisir de visualiser l'information selon le type des éléments tracés. Cela permet par exemple de voir chaque mois la quantité de consultations du cours, de réponses aux exercices ou de voir quels apprenants ont été actifs chaque mois.

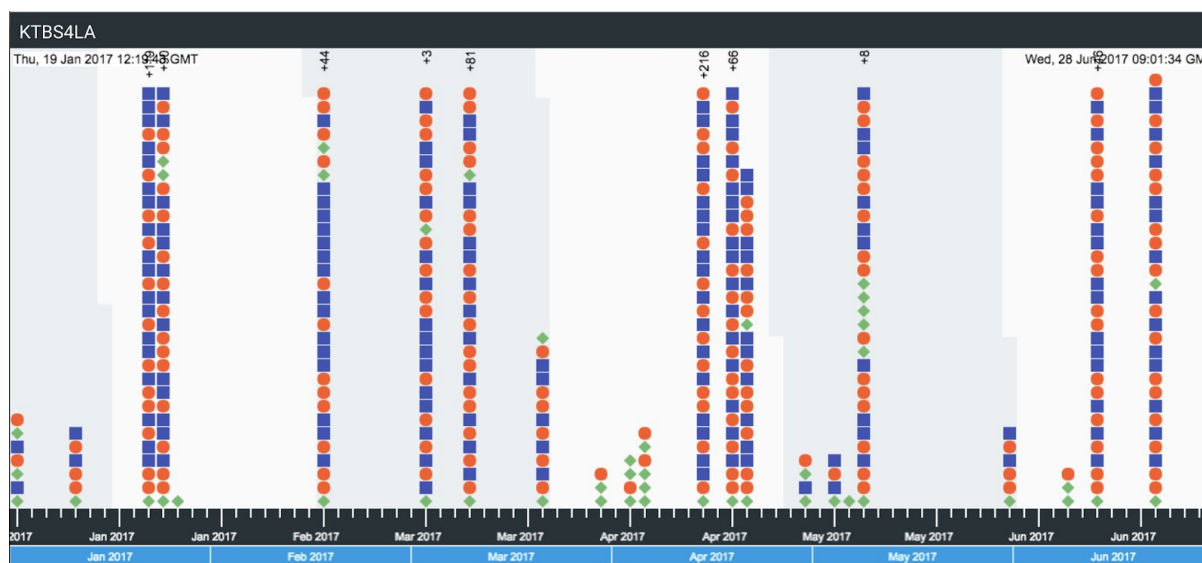


Figure 32. KTBS4LA - Visualisation de traces dans une time line

Une visualisation sous forme de *time line* permet de réaliser une analyse plus poussée en exploitant la temporalité des différents éléments observés. Sans compétence en programmation, l'enseignant peut définir des règles décrivant quelle forme et quelle couleur utiliser pour représenter un élément de la trace vérifiant certaines contraintes. Par exemple, il pourra choisir de faire disparaître les éléments de type *consultation d'un cours*, faire apparaître comme un losange vert les éléments de type *demande d'aide*, comme un carré bleu les éléments de type *réponse à un exercice* dont l'attribut *niveau de difficulté* est supérieur à un seuil et comme des ronds rouges ceux dont l'attribut *niveau de difficulté* est inférieur à ce seuil. Cet exemple de visualisation peut permettre à l'utilisateur de voir si les apprenants demandent davantage d'aide dans les exercices difficiles que dans les exercices plus faciles.

D3KODE

D3KODE est une application Web de type client léger conçu en 2012 dans le cadre de la thèse d'Olivier Champalle, dirigée par Alain Mille et Karim Sehaba. Cette application permet de stocker, de transformer et de visualiser plusieurs niveaux de traces numériques sur la base du concept de « trace modélisée » développé par l'équipe SILEX du LIRIS.

Les principaux enjeux de D3KODE reposent sur plusieurs objectifs complémentaires :

- procurer les moyens nécessaires pour exploiter/analyser tout type de données numériques modélisées suivant les principes de la trace modélisée ;
- faciliter l'inscription des connaissances d'exploitation/d'analyse dans un but de partage et de réutilisation et ce auprès d'un large public non informaticien.

Les modèles de traces et de transformations représentent des connaissances d'analyse qui peuvent être ainsi réutilisées par d'autres utilisateurs pour abstraire des « observables » de plus hauts niveaux à partir de nouvelles traces de même modèle.

D3KODE possède plusieurs fonctionnalités permettant aux utilisateurs en fonction de leur profil, administrateur, expert ou stagiaire, de créer facilement des modèles de traces, des transformations, d'importer des traces, d'exécuter des transformations et de visualiser les résultats (Figure 33).

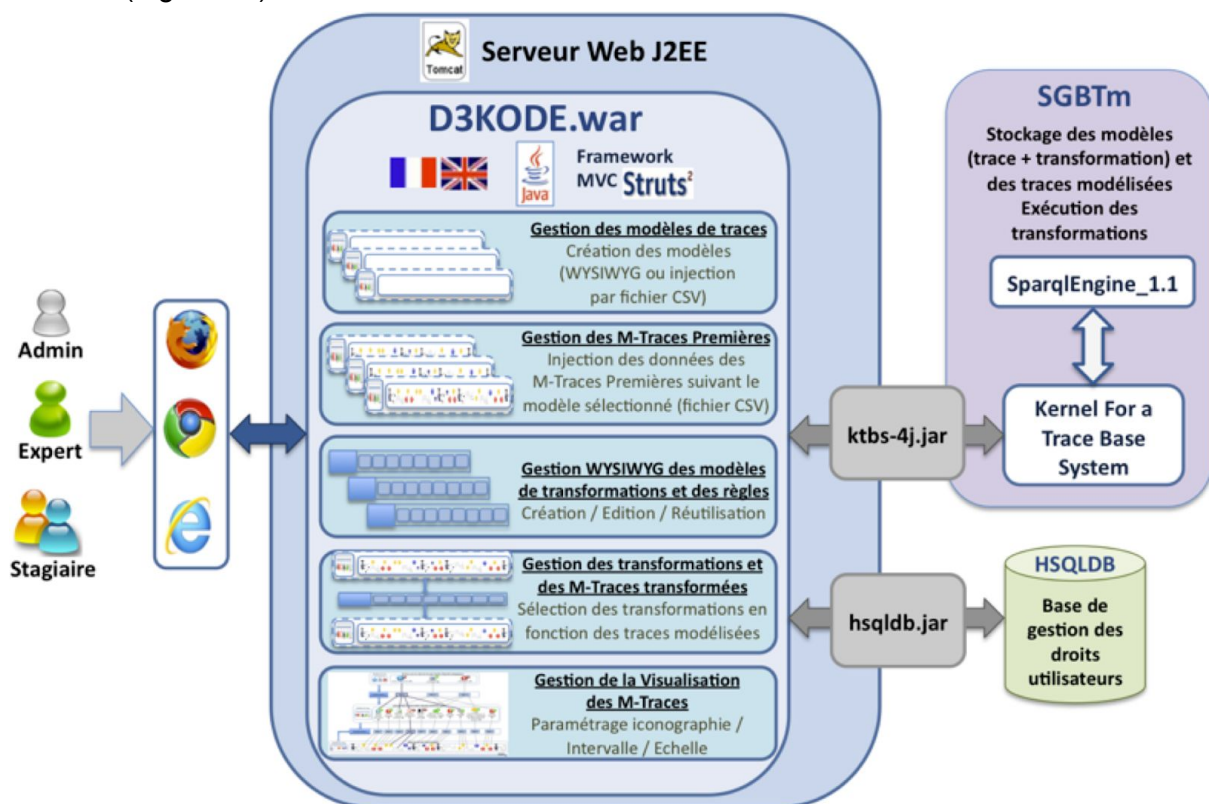


Figure 33. Architecture et fonctionnalités de D3KODE

D3KODE utilise la toute première version du KTBS qui n'est aujourd'hui plus en activité. Cette version a été augmentée pour les besoins de D3KODE. En effet la version du KTBS employée pour D3KODE était basée sur l'emploi de SPARQL 1.0 pour requêter les fichier RDF. Les besoins de D3KODE nécessitaient du SPARQL 1.1 notamment pour des observations de « non-présence » d'informations.

L'interface de D3KODE s'adresse à des utilisateurs non informaticiens et a donc été conçu de manière à faciliter l'analyse de traces numériques en dégageant l'utilisateur des notions techniques du KTBS et en particulier du langage de requête SPARQL. Plusieurs interfaces permettent aux utilisateurs de créer « graphiquement » des modèles de trace (Figure 34), de transformations, d'importer des traces et d'exécuter les transformations (Figure 35).

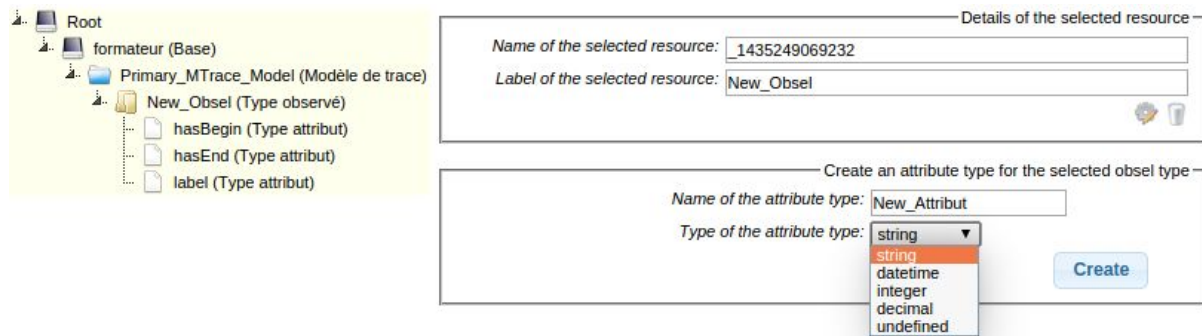


Figure 34. Interface de création de modèle de trace

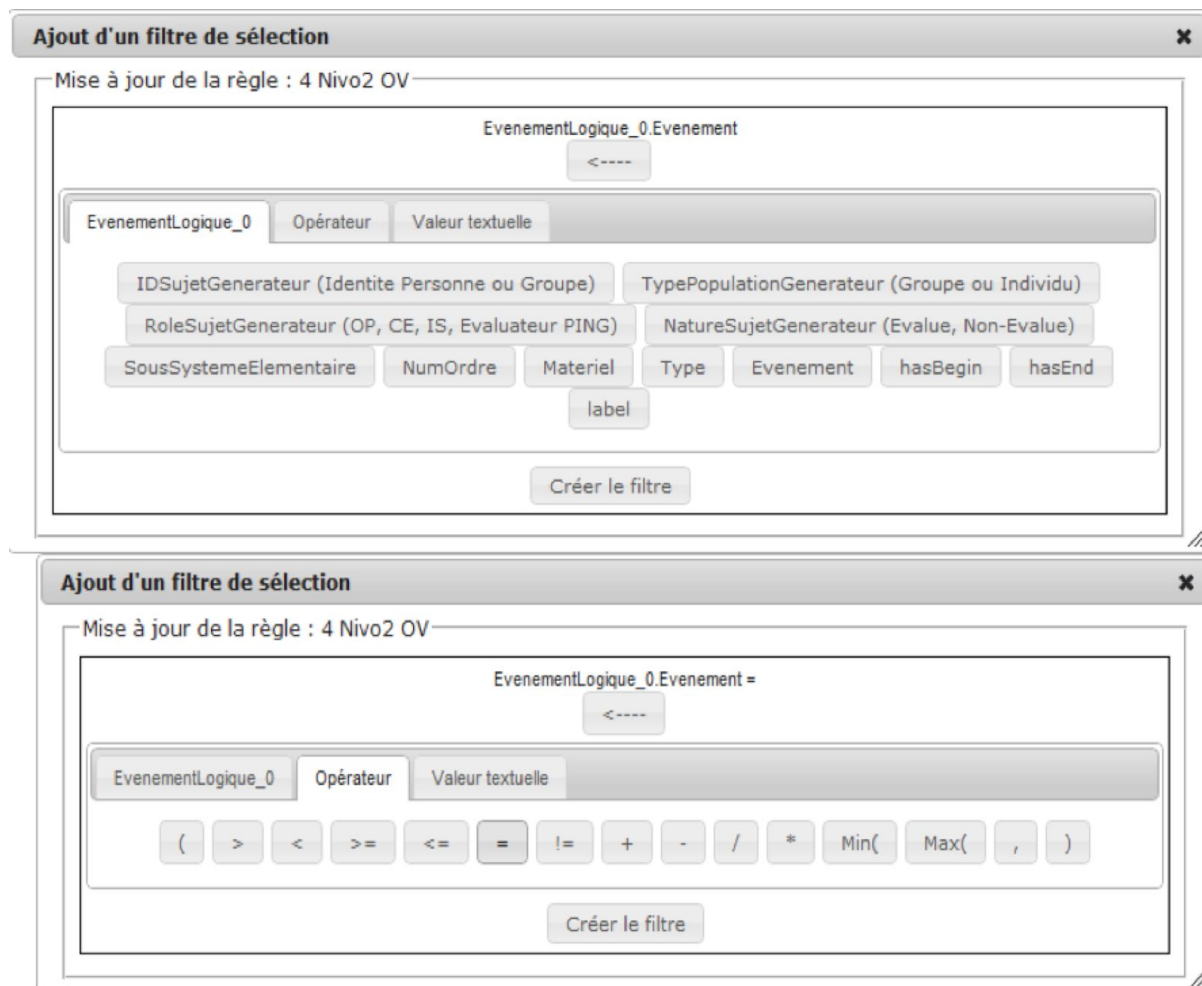


Figure 35. Interface de création d'une règle de transformation

Afin de faciliter l'analyse, la synthèse visuelle est interactive (Figure 35 et 36). Il est ainsi possible pour un utilisateur de « cliquer » sur une transformation et d'accéder à son descriptif afin de comprendre quelles sont les connaissances qui ont été mobilisées pour construire un observable de plus haut niveau. Chaque observable possède aussi un lien

visuel sur ses origines. De cette manière les utilisateurs peuvent explorer les différents niveaux de l'activité et mieux en comprendre le sens.

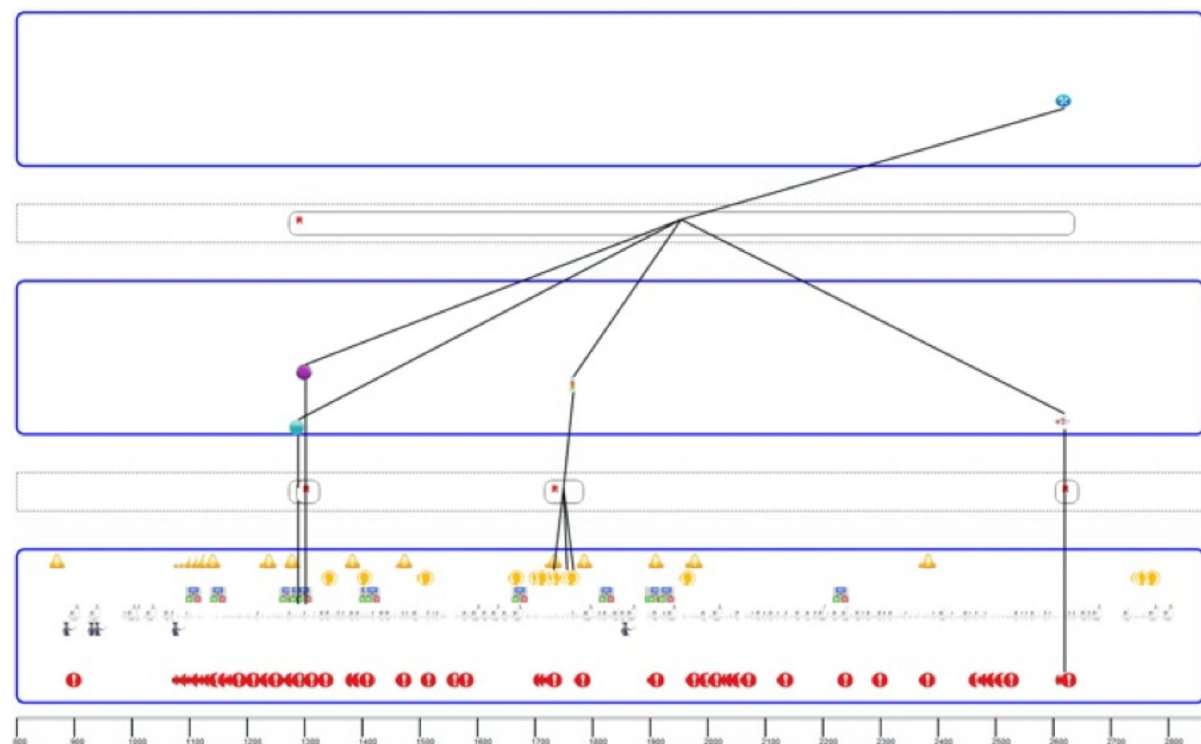


Figure 36. Exemple de synthèse visuelle sur 3 niveaux

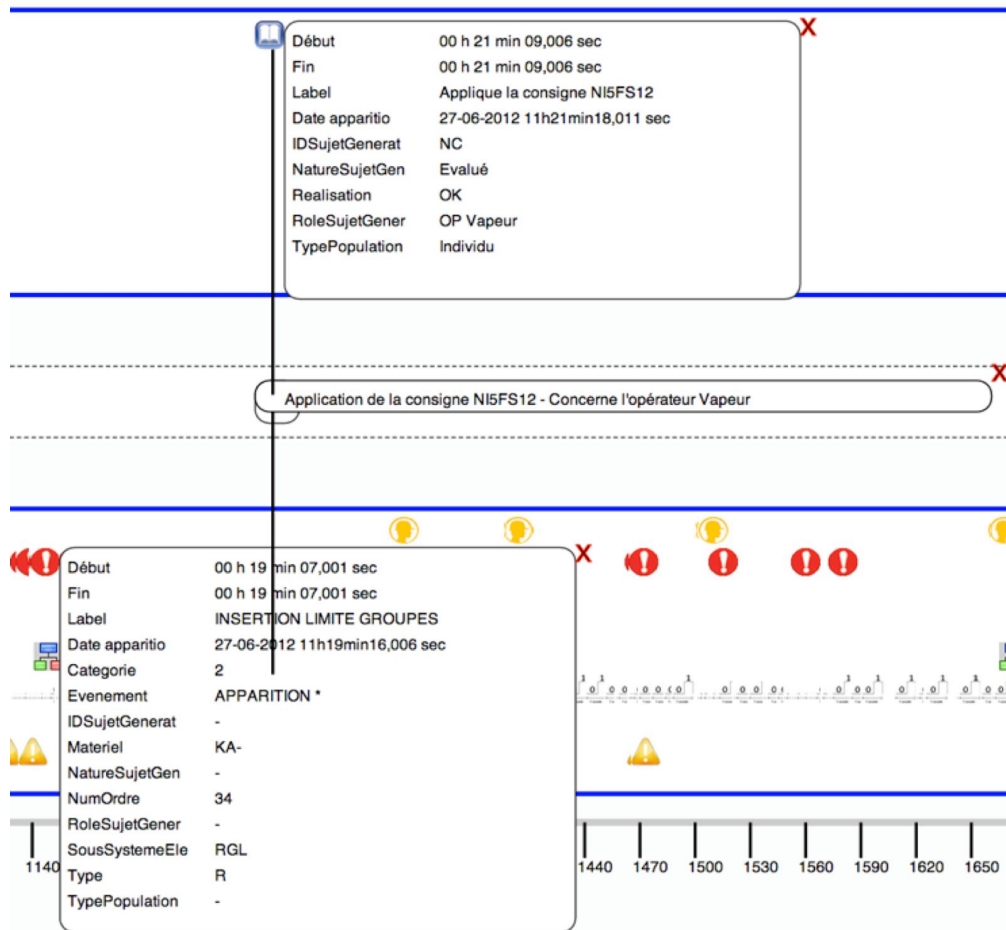


Figure 37. Exploration interactive des traces de la synthèse visuelle

EMODA

Le tableau de bord EMODA [Ez-Zaouia et al., 2017] est un outil conçu pour aider les enseignants engagés dans des formations en ligne pour l'apprentissage des langues. Il permet de capturer, analyser et visualiser les émotions des apprenants lorsque des activités de communication synchrones sont proposées avec des outils de visio-conférence et audio-conférence. Le fait de visualiser les émotions de l'apprenant permet aux enseignants de créer et maintenir des relations socio-affective lors de la formation.

EMODA combine des modèles d'émotions discrets et dimensionnels. Les modèles discrets prennent en compte un ensemble fini d'émotions (colère, peur, joie, tristesse, surprise, ...) considérées comme universelles et qui constituent le socle des émotions humaines. Les modèles dimensionnels combinent plusieurs aspects qualifiants les émotions. Dans EMODA sont considérés la valence qui est la valeur positive ou négative d'une émotion et l'excitation qui est la manifestation physique et l'intensité d'une émotion (variant de faible à forte). EMODA propose des résultats sous forme d'indicateurs discrets, bi ou multidimensionnels. Différents formats de visualisation sont proposés : histogramme, émoticône, courbes d'évolution temporelle, images significatives de la séance.

Présentation générale du tableau de bord

Le tableau de bord est composé de 4 parties. Une barre d'outils, proposée tout en haut du tableau de bord permet de à l'enseignant de naviguer entre ses sessions et d'avoir une vue synthétique des émotions les plus caractéristique des sessions.

Dans la partie supérieure les indicateurs donnent des informations globales sur l'activité et les émotions ressentie. Dans la partie intermédiaire des informations sur le contexte de l'activité sont proposés, l'enseignant peut revoir ou réentendre certains moments de la session. Dans la partie inférieure différentes courbes représentant les évolutions au cours du temps des émotions et des activités permettent d'affiner l'analyse et de comprendre les raisons qui construisent les émotions significatives.

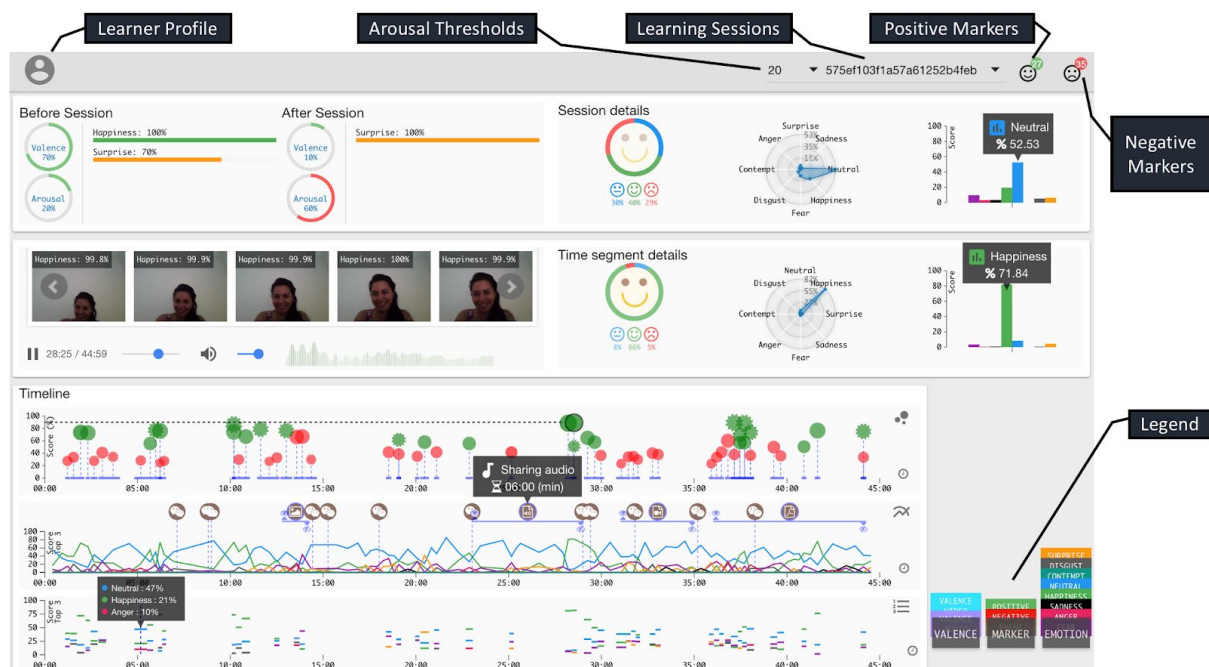


Figure 38. EMODA - Interface du tableau de bord

Architecture de traitement des données

EMODA adopte une approche multimodale et considère quatre sources de données: audio, vidéo, self-report, et traces d'interaction.

Les données audio et vidéo correspondent aux échanges de communication enregistrés lors des séances. Les données de self-report sont renseignés par l'apprenant avant et après la séance. Les traces d'interactions sont utilisées pour qualifier les actions liées à des émotions particulières au cours de l'activité.

Différentes APIs ont été combinées. L'ensemble du processus de traitement est présenté dans la figure 39.

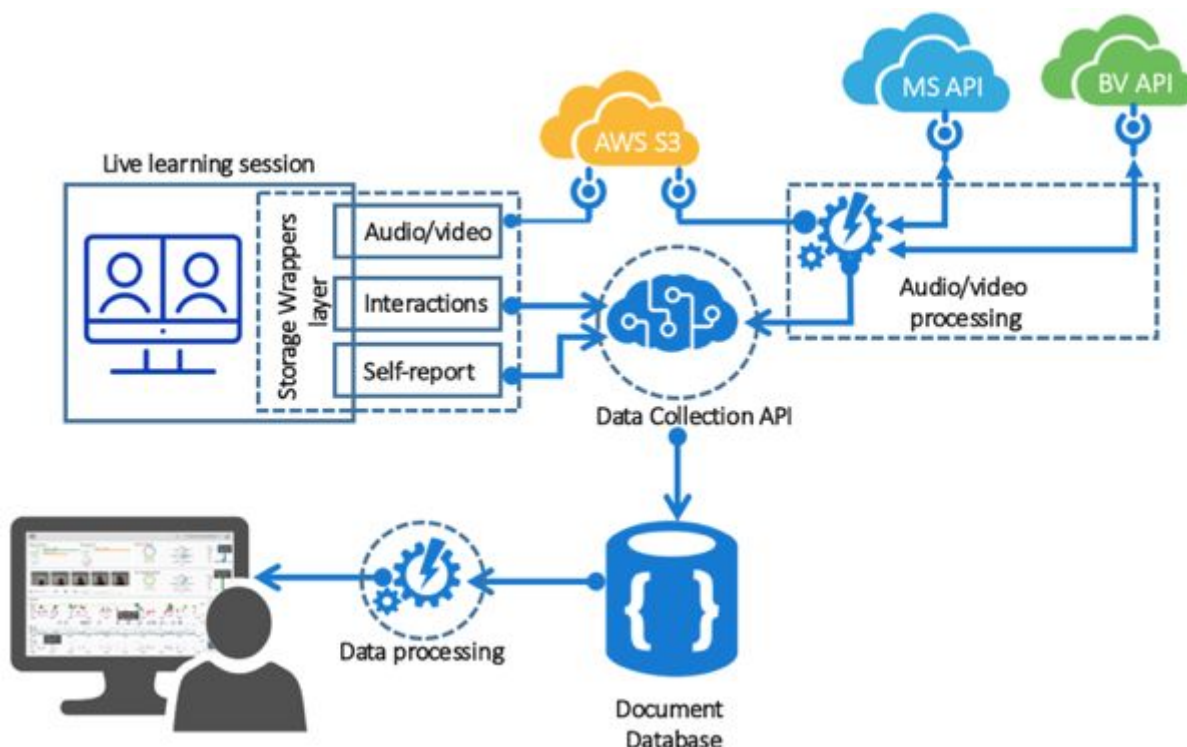


Figure 39. EMODA - Architecture générale

Les sessions d'apprentissage (audiovisuelles) sont enregistrées dans le cloud en utilisant AWS S3. L'API *Microsoft emotion recognition Service* (MS API) est utilisée pour caractériser les émotions selon le modèle discret à partir des expressions faciales (vidéo). L'API *Beyond Verbal service* (BV API) caractérise les émotions selon le modèle dimensionnel valence-excitation à partir de la voix (audio).

Les émotions caractérisées sont combinées aux émotions auto-rapportées et aux traces d'interaction dans une base de données grâce à une API spécifique à l'application. Un module de traitement des données permet de produire les indicateurs présents dans le tableau de bord.

Une étude pilote a été menée et a montré que l'outil et les principes de visualisation choisis étaient globalement pertinents pour les enseignants. Cette étude a montré aussi que l'exploitation de différentes API, externes et gratuites, était possible pour faire les traitements post-activité.

Perspectives

Les travaux menés dans le cadre des Learning Analytics sont assez variés et couvrent toute la chaîne de traitement des données. Toutefois de nombreuses étapes restent encore à mettre en œuvre afin notamment de diffuser, d'exploiter et d'étendre ces résultats. En effet, les travaux menés dans le cadre du projet ANR HUBBLE ont cherché à aider la capitalisation et le partage des processus d'analyse. Au travers de leur outil, il est possible de décrire les processus et aussi d'obtenir toutes les informations afin de réutiliser ceux déjà saisis. Un autre enjeu lié à ce dernier, consiste à travailler sur l'adaptation des processus d'analyse en fonction des différentes plateformes et des formats de données utilisés. Les

étapes de prétraitement et d'intégration étant coûteuses, il est nécessaire de trouver des solutions facilitant cette réutilisation.

La production de Learning Analytics soulève les questions liées à l'exploitation de ces informations. Le premier usage se porte sur des tableaux de bord et donc des retours graphiques et visuels, mais l'intégration dans les outils pédagogiques sous la forme de tuteurs intelligents ou d'adaptation des plateformes est aussi primordiale. Le retour en force de l'intelligence artificielle permet l'exploitation de données massives en couplant les Learning Analytics et le domaine du Data Mining (Educational Data Mining), ce qui amène les chercheurs à travailler sur l'exploitation automatique mais aussi l'explicabilité des processus et des données obtenues afin d'améliorer l'appropriation par les usagers (enseignants, apprenants, ...).

Enfin, se pose aussi la question des entrepôts de données dédiés à l'éducation dans lesquels les chercheurs pourraient éprouver leurs théories ainsi que les outils développés. Ces entrepôts seraient aussi importants pour les usagers eux mêmes qui pourraient déposer leurs données et bénéficier des outils de la recherche adaptés et directement disponibles dans ces entrepôts.

Références

- [Aleven et al., 2006]** Aleven, V., McLaren, B., Roll, I. Koedinger, K.: Toward meta-cognitive tutoring: A model of help seeking with a Cognitive Tutor. *International Journal of Artificial Intelligence in Education*, Vol. 16 N. 2, p. 101–128 (2006).
- [Barazzutti et al., 2016]** Barazzutti, P.-L., Cordier, A., Fuchs, Béatrice., Crémilleux, Bruno., Runz, Cyril de. : Transmute : un outil interactif pour assister l'extraction de connaissances à partir de traces. *Extraction et Gestion des Connaissances (EGC 2016)*, pages 463-468
- [Bienkowski et al., 2012]** M. Bienkowski, J. Brecht, and J. Klo. The learning registry: building a foundation for learning resource analytics. In *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge (LAK 12)*, pages 208–211. ACM, 2012.
- [Blomer, 2012]** Blomer, J. (2012). NsdI paradata.
https://wiki.ucar.edu/display/nsdIdocs/comm_para. Accédé le 3 octobre 2018.
- [Bouhineau et al., 2013]** Bouhineau, D., Luengo, V., Mandran, N., Ortega, M., Wajeman, C. (2013) Open platform to model and capture experimental data in Technology enhanced learning systems. *Workshop Data Analysis and Interpretation for Learning Environments*.
- [Broisin et al., 2017a]** Broisin, J., Venant, R., Vidal, P.: Lab4CE: a Remote Laboratory for Computer Education. *Int. Journal of Artificial Intelligence in Education*, Springer, Vol. 27 N. 1, p. 154-180 (2017).
- [Broisin et al., 2017b]** Broisin, J., Venant, R., Vidal, P.: Awareness and Reflection in Virtual and Remote Laboratories: the case of Computer Education. *Int. Journal of Technology Enhanced Learning*, Inderscience Publishers, Vol. 9 N. 2/3, p. 254-276 (2017).
- [Casado et al., 2017]** kTBS4LA : Casado, R., Guin, N., Champin, P.-A., Lefevre, M.. kTBS4LA : une plateforme d'analyse de traces fondée sur une modélisation sémantique des traces. *Méthodologies et outils pour le recueil, l'analyse et la visualisation des traces d'interaction - ORPHEE-RDV*, Font-Romeu, hal-01637548, 2017.

- [Champalle et al., 2016]** Champalle, O., Sehaba, K., Mille, A.: Observation et analyse de comportements des utilisateurs à base de traces. *Revue des Sciences et Technologies de l'Information - Série TSI : Technique et Science Informatiques*, Lavoisier, 2016, 35 (4-5/2016)
- [Champin et al., 2013]** KTBS : Champin, P.-A., Mille, A., Prié, Y.. Vers des traces numériques comme objets informatiques de premier niveau : une approche par les traces modélisées. *Revue Intellectica*, pp.171-204, hal-00924203, 2013.
- [Clow, 2012]** Cycle-2. Clow, D., "The learning analytics cycle: closing the loop effectively," in *Proceedings of the LAK 2012*, pp. 134–138, ACM, 2012.
- [Dabbebi et al., 2017]** Dabbebi, I., Iksal, S., Gilliot, J.-M., May, M., Garlatti, S.. Towards Adaptive Dashboards for Learning Analytic: An Approach for Conceptual Design and implementation. 9th International Conference on Computer Supported Education (CSEDU 2017), Apr 2017, Porto, Portugal. *Proceedings of the 9th International Conference on Computer Supported Education*, pp.120-131, 2017
- [Derbel et al., 2015]** Derbel, F., Champin, P. A., Cordier, A., & Munch, D. (2015, June). Authentification d'un utilisateur à partir de ses traces d'interaction. In *Treizièmes Rencontres des Jeunes Chercheurs en Intelligence Artificielle (RJCIA 2015)*.
- [Djouad et al., 2011]** T. Djouad, A. Mille, M. Benmohammed. " SBT-IM: Système à base de traces-Indicateurs d'interactions Moodle". Conférence EIAH 2011, Mons, 2011.
- [Dyke et al., 2010]** G. Dyke, K. Lund, J-J. Girardot. "Tatiana : un environnement d'aide à l'analyse de traces d'interactions humaines". *Technique et science Informatiques (TSI)*, pp 1179-1205 (2010).
- [Ez-Zaouia et al., 2017]** Ez-Zaouia M., Lavoué E.. EMODA: a tutor oriented multimodal and contextual emotional dashboard. *Proceedings of the Seventh International Learning Analytics Knowledge Conference (LAK 2017)*, Mar 2017, Vancouver, Canada, pp.429-438 (2017)
- [Fayyad et al., 1996]** Cycle-1. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., "From data mining to knowledge discovery in databases," *AI magazine*, vol. 17, no. 3, p. 37, 1996.
- [Fuchs, 2018]** Fuchs, B.. Focaliser l'extraction d'épisodes séquentiels à partir de traces par le contexte. In *29es Journées Francophones d'Ingénierie des Connaissances, IC 2018* (pp. 213-227) (2018, July).
- [Gee, 2007]** Gee, J.P. What Video Games Have to Teach Us About Learning and Literacy. Second Edition: Revised and Updated Edition (2nd ed.). (2007). Palgrave Macmillan.
- [Georgeon et al., 2012]** Georgeon, O. L., Mille, A., Bellet, T., Mathern, B., Ritter, F. E. : Supporting Activity Modeling from Activity Traces. *Expert Systems*, 29 (3), 261-275, 2012
- [Greller et al., 2012]** Actor - Greller, W., Drachsler, H.. "Translating learning into numbers: A generic framework for learning analytics.," *Journal of Educational Technology & Society*, vol. 15, no. 3, pp. 42–57, 2012.
- [Gutierrez-Santos et al., 2010]** Gutierrez-Santos, S., Mavrikis, M., Magoulas, G.: Layered development and evaluation for intelligent support in exploratory environments: the case of microworlds. *Intelligent Tutoring Systems (ITS 2010)*, p. 105-114 (2010).
- [Hecking et al., 2014]** Hecking, T., Manske, S., Bollen, L., Govaerts, S., Vozniuk, A., Hoppe, H. U.: A flexible and extendable learning analytics infrastructure. *Advances in Web-Based Learning (ICWL 2014)*, p. 123-132 (2014).
- [Iksal, 2011]** Iksal, S.. Tracks Analysis in Learning Systems: A prescriptive Approach. *International Journal for e-Learning Security (IJeLS)*, 2011, pp.3--9.

- [Iksal, 2012] Iksal, S.. Ingénierie de l'observation basée sur la prescription en EIAH. HDR en Informatique. Université du Maine, 2012.
- [Kong et al., 2015] Chang, B. K. W., Lefevre, M., Guin, N., & Champin, P. A. (2015, June). SPARE-LNC: un langage naturel contrôlé pour l'interrogation de traces d'interactions stockées dans une base RDF. In IC2015.
- [Lesbard et al., 2006] Lesnard, L. et de Saint Pol, T.. Introduction aux méthodes d'appariement optimal. Bulletin de Méthodologie Sociologique, n° 90, pp. 5-25, 2006.
- [LTSC, 2002] Learning Technology Standards Committee. (2002). IEEE Standard for learning object metadata. IEEE Standard, 1484(1), 2007-04.
- [Long et al., 2011] Long, P. D. et Siemens G. (2011). Penetrating the Fog: Analytics in Learning and Education. Educause Review, 46(5), 30-40. Récupéré de <https://net.educause.edu/ir/library/pdf/ERM1151.pdf>
- [Loup et al., 2016] Loup, G., Serna, A., Iksal, S., George, S.. Immersion and Persistence: Improving Learners' Engagement in Authentic Learning Situations. 11th European Conference on Technology Enhanced Learning, EC-TEL 2016, Sep 2016, Lyon, France. pp.410-415, Adaptive and Adaptable Learning.
- [Lukarov et al., 2014] Lukarov, V., Chatti, M. A., Thüs, H., Kia, F. S., Muslim, A., Greven, C., & Schroeder, U. (2014). Data Models in Learning Analytics. In DeLFI Workshops, pp. 88-95.
- [Mannila et al., 1997] Mannila, H., Toivonen, H., & Verkamo, A. I. (1997). Discovery of frequent episodes in event sequences. *Data mining and knowledge discovery*, 1(3), 259-289.
- [May et al., 2011] M. May, S. George, P. Prévôt. *TrAVis to Enhance Online Tutoring and Learning Activities: Real Time Visualization of Students Tracking Data*. International Journal of Interactive Technology and Smart Education (ITSE), vol. 8(1), pp. 52-69, 2011.
- [Michel et al., 2017] Michel, C., Lavoué, E., George, S., Ji, M.. Supporting Awareness and Self-Regulation. In Project-Based Learning through Personalized Dashboards, publication soumise à International Journal of Technology Enhanced Learning (IJTEL), vol. 9, 2/3, pp. 204-226, 2017.
- [Muratet et al., 2016] Muratet, M., Yessad, A., Carron, T.. Understanding Learners' Behaviors in Serious Games. ICWL 2016 - International Conference on Web-based Learning, 2016
- [Muslim et al., 2016] Muslim, A., Chatti, M. A., Mahapatra, T., & Schroeder, U. (2016). A rule-based indicator definition tool for personalized learning analytics. In Proceedings of the Sixth International Conference on Learning Analytics & Knowledge, pp. 264-273.
- [Niemann et al., 2012] K. Niemann, M. Scheffel, and M. Wolpers. An overview of usage data formats for recommendations in TEL. In Workshop on Recommender Systems for Technology Enhanced Learning (RecSysTEL 12), pages 95–100, 2012.
- [Pham Thi Ngoc, 2011] Pham Thi Ngoc, D., Spécification et conception de services d'analyse de l'utilisation d'un environnement informatique pour l'apprentissage humain. Thèse de doctorat en Informatique, Université du Maine, 2011.
- [Sehaba et al., 2017] Sehaba, K., Champalle, O., Mille, A. : Assisting activity analysis in professional learning environments. Case study: activity analysis in professional learning environments. Case study: activity analysis of trainees on nuclear power plant full-scale simulators. International Journal of Learning Technology, vol. 12(2), pp. 88-118, hal-01639886, 2017.

- [Snell et al., 2012a]** Snell, J., Atkins, M., Norris, W., Messina, C., Wilkinson, M., Dolin, R. 2012. JSON Activity Streams 1.0. Accédé le 4 octobre 2018. <http://activitystrea.ms/specs/json/1.0/>
- [Snell et al., 2012b]** Snell, J., Atkins, M., Recordon, D., Messina, C., Keller, M., Steinberg, A., Dolin, R. 2012. Activity Base Schema (Draft). Accédé le 4 octobre 2018. <http://activitystrea.ms/specs/json/schema/activity-schema.html>
- [Stamper et al., 2011]** Cycle-3. J.C. Stamper, K.R. Koedinger, R.S.J.d. Baker, A. Skogsholm, B. Leber, S. Demi, S. Yu, and D. Spencer, "Managing the educational dataset lifecycle with datashop," in Proceedings of the AIED 2011 (G. Biswas, S. Bull, J. Kay, and A. Mitrovic, eds.), (Berlin, Heidelberg), pp. 557–559, Springer Berlin Heidelberg, 2011.
- [Thomas et al., 2011]** Thomas, P., Yessad, A. et Labat, J.-M. Petri nets and ontologies: Tools for the "learning player" assessment in serious games. In ICALT, 415-419. (2011).
- [Venant et al., 2016]** Venant, R., Vidal, P., Broisin, J.: Evaluation of Learner Performance during Practical Activities: an Experimentation in Computer Education. IEEE International Conference on Advanced Learning Technologies (ICALT 2016), IEEE Computer Society, p. 237-241 (2016).
- [Venant et al., 2017]** Venant, R., Sharma, K., Vidal, P., Dillenbourg, P., Broisin, J.: Etude du Comportement des Apprenants en Situation de Travaux Pratiques et de son Influence sur leur Réussite Académique. Environnement Informatique pour l'Apprentissage Humain (EIAH 2017), Association des Technologies de l'Information pour l'Education et la Formation (ATIEF), p. 17-28 (2017).
- [Venant et al., 2017b]** Venant, R., Sharma, K., Vidal, P., Dillenbourg, P., Broisin, J.: Using sequential pattern mining to explore learners' behaviors and evaluate their correlation with performance in inquiry-based learning. European Conference on Technology Enhanced Learning (EC-TEL 2017), Springer, p. 286-299 (2017).
- [Wolpers et al., 2007]** M. Wolpers, J. Najjar, K. Verbert, and E. Duval. Tracking actual usage: the attention meta- data approach. In International Journal of Educational Technology and Society, pages 1176–3647. Press, 2007.
- [Yessad et al., 2017]** Yessad, A., Muratet, M., Carron, T.. Aider à l'analyse du comportement d'un apprenant dans les jeux sérieux. EIAH 2017, Strasbourg, France (2017)
- [Young, 2009]** Young, H. P. Learning by trial and error. Games and economic behavior, 65(2), 626-643. (2009).
- [Zarka et al., 2013]** Zarka, R., Champin P.-A., Cordier, A., Egyed-Zsigmond, E., Lamontagne, L., Mille, A. (2013). « TStore: A Trace-Base Management System using Finite-State Transducer Approach for Trace Transformation ». International Conference on Model-Driven Engineering and Software Development (MODELSWARD 2013), 19 février 2013, Barcelona (Espagne), pp. 117-122. HAL : hal-01351699.

Annexe 1 : Liste des outils étudiés

Abstract (LIRIS) : [Georgeon et al., 2012]
Activity Streams : [Snell et al., 2012a]
Contextualized Attention Metadata (Fraunhofer FIT) : [Wolpers et al., 2007]
CSVtoAPI - xAPI Lab (Projet ANR Kolflow)
D3KODE (LIRIS) : [Champalle et al., 2016]
DDART+Reporting tool (LIRIS) : [Michel, et al, 2017]
DisKit (DIScovering Knowledge from Interaction Traces) (LIRIS) : [Fuchs, 2018]
dmt4sp (LIRIS) : [Mannila et al., 1997]
EMODA (LIRIS) : [Ez-Zaouia et al., 2017]
IMS Caliper
kTBS (a kernel for Trace-Based Systems) (LIRIS) : [Champin et al., 2013]
kTBS4LA (LIRIS) : [Casado et al., 2017]
Laalys (Université Pierre et Marie Curie) : [Muratet et al., 2016] [Yessad et al., 2017]
Lab4ce (IRIT) : [Broisin et al., 2017a] [Broisin et al., 2017b]
LCDM (Learning Technologies)
LEA4AP (Université Pierre et Marie Curie)
Learning Registry (SRI International)
Limesurvey
Méthode statistique de recherche de profils / typologie à partir de descripteurs (proposée dans les logiciels d'analyse statistique)
Méthodes statistiques d'appariement optimal - analyse de séquences (proposée dans les logiciels d'analyse statistique)
NSDL Paradata (Fraunhofer Institute for Applied Information Technology) : [Niemann et al., 2012]
Samotraces (LIRIS) : Site du projet SAMOTRACES : <http://sourceforge.net/projects/samotraces/>
SamoTraceMe (LIRIS) : [Derbel et al., 2015]
SBT-IM (LIRIS) : [Djouad et al., 2011]
SPARE LNC (LIRIS) : [Kong et al., 2015]
T-store (LIRIS) : [Zarka et al., 2013]
Taaabs (LIRIS) : Site du projet TAAABS :
<https://projet.liris.cnrs.fr/sbt-dev/tbs/doku.php/tools:taaabs>
Tactiléo Map
TraceMe (LIRIS)
Transmute (LIRIS) : [Barazzutti et al., 2016]
Tatiana (ICAR) : [Dyke et al., 2010]
TRAVIS (LIRIS-LIUM) : [May et al., 2011]
UnderTracks (LIG) : [Bouhineau et al., 2013]
UTL (LIUM) : [Iksal, 2012][Iksal, 2011][Loup et al., 2016][Dabbebi et al., 2017][Pham Thi Ngoc, 2011]
xCollector (LIRIS)