



HAL
open science

Evaluation et visualisation de la performance de l'apprenant lors d'une activité pratique

Rémi Venant

► **To cite this version:**

Rémi Venant. Evaluation et visualisation de la performance de l'apprenant lors d'une activité pratique. 6emes Rencontres Jeunes Chercheurs en Environnements Informatiques pour l'Apprentissage Humain (RJC-EIAH 2016), Jun 2016, Montpellier, France. pp. 39-44. hal-01737044

HAL Id: hal-01737044

<https://hal.science/hal-01737044>

Submitted on 19 Mar 2018

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.



Open Archive TOULOUSE Archive Ouverte (OATAO)

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible.

This is an author-deposited version published in : <http://oatao.univ-toulouse.fr/>
Eprints ID : 19182

The contribution was presented at RJC-EIAH 2016 :
<https://rjceiah.sciencesconf.org/>

To cite this version : Venant, Rémi *Evaluation et visualisation de la performance de l'apprenant lors d'une activité pratique*. (2016) In: 6emes Rencontres Jeunes Chercheurs en Environnements Informatiques pour l'Apprentissage Humain (RJC-EIAH 2016), 16 June 2016 - 17 June 2016 (Montpellier, France).

Any correspondence concerning this service should be sent to the repository administrator: staff-oatao@listes-diff.inp-toulouse.fr

Evaluation et visualisation de la performance de l'apprenant lors d'une activité pratique

Rémi Venant
2nd année

Univ. Toulouse III, Institut de Recherche en Informatique de Toulouse (IRIT), 115 route de Narbonne 31400 Toulouse
Remi.venant@irit.fr

Résumé

La prise de conscience de l'apprenant sur la performance de son activité ainsi que celle de ses pairs fait l'objet de nombreuses recherches. Dans le cadre des activités pratiques à distance, cette awareness personnelle et sociale doit être rendue tout au long de la session d'apprentissage. Cette problématique est abordée ici en trois étapes : (1) la définition d'une métrique de performance, (2) la collecte et l'analyse de données pour inférer cette métrique en temps réel, et (3) la restitution de la performance aux apprenants, au cours de leur activité pratique, sans augmenter leur charge cognitive. Nous proposons dans cet article (i) une métrique pour notre contexte de l'enseignement de l'informatique, (ii) un framework générique pour l'analyse de données et l'inférence d'indicateurs, et (iii) des outils de visualisation. Nos propositions ont été implantées dans Lab4CE, un environnement de télé-TPs pour l'informatique, et expérimentées dans un contexte d'apprentissage réel. Si les outils sont utilisés par la plupart des apprenants, l'expérimentation montre l'existence d'une relation entre leur utilisation et la performance, ainsi qu'une tendance à leur exploitation pour la recherche d'aide exécutive.

Introduction

Les Travaux Pratiques à distance (télé-TPs) s'appuient sur l'apprentissage exploratoire qui vise, entre autres, à l'apprentissage profond et à la réflexion (de Jong et al. 2013). Avec ces enseignements, les apprenants doivent s'approprier le savoir théorique acquis préalablement par la manipulation et l'analyse de ressources distantes, réelles ou virtuelles. Les télé-TPs sont mis en œuvre à travers des systèmes complexes qui soulèvent différentes problématiques matérielles et logicielles. Les recherches dans ce domaine concernent essentiellement la conception d'architectures (Harward et al. 2008), le partage de ressources (Lowe et al. 2013), la répartition de charge (San Cristobal et al. 2010) et la standardisation (Salzmann et al. 2015). Toutefois, la présence d'artefacts pédagogique a été montré comme important pour l'engagement de l'apprenant dans ces activités (Corter et al. 2011).

Au sein des environnements d'apprentissage traditionnels, l'(auto-)évaluation de la performance des apprenants représente une information critique qui doit leur être fournie (Arnold & Pistilli 2012). Dans cette optique, une approche commune est l'utilisation d'outils d'analyse de données d'apprentissage

(*learning analytics* en anglais) et d'outils d'awareness pour concevoir des tableaux de bord qui fournissent un retour sur les résultats généraux des apprenants (Prensky 2011), leur niveau global de performance ou encore leurs forces et faiblesses (Howlin & Lynch 2014). Pour cela, ces outils s'appuient sur l'évaluation de l'acquisition de connaissances par les apprenants et de leur maîtrise de concepts théoriques. Cependant, dans ce contexte d'activités pratiques que représentent les télé-TPs, ces techniques sont inappropriées car elles ne permettent pas de mesurer la capacité des apprenants à appliquer leurs connaissances théoriques pendant une situation d'apprentissage pratique.

Nous étudions donc dans cet article la question suivante : comment doter un environnement de télé-TPs d'outils d'awareness personnels et sociaux pour refléter aux apprenants leurs niveaux de performance ? Les travaux présentés ont été réalisés dans le contexte de Lab4CE, un EIAH (environnement informatique pour l'apprentissage humain) pour l'enseignement pratique de l'informatique (Broisin et al. 2015). Notre approche repose sur les processus de récolte et d'analyse de données d'apprentissage pour tracer et étudier les interactions entre les apprenants et leurs ressources, inférer des indicateurs pédagogiques, et les exposer via différentes techniques de visualisation.

La section suivante identifie les besoins pour évaluer la performance des apprenants au sein d'activités pratiques. La troisième section présente Lab4CE, notamment ses fonctionnalités d'analyse de données pour le calcul d'indicateurs pédagogiques, ainsi qu'une métrique de performance en accord avec nos contextes technique et pédagogique. Celle-ci est ensuite exploitée par deux outils d'awareness pour rapporter le niveau de performance des apprenants pendant leurs activités pratiques, et permettre l'étude de leurs actions. Dans une dernière section, nous présentons une expérimentation menée dans un contexte réel d'apprentissage pour évaluer nos contributions.

Besoins pour l'évaluation dans les Télé-TPs

La performance de l'apprenant illustre sa capacité à démontrer des aptitudes spécifiques et la maîtrise de connaissances (Achumba et al. 2013). Sa mesure repose classiquement sur des évaluations effectuées après l'activité d'apprentissage (Arnold & Pistilli 2012), comme proposées par (Ritzhaupt et al. 2015), ou par l'observation physique (Achumba et al. 2013).

Par nature, la mesure a posteriori est incompatible

avec toute forme d'awareness au cours d'une activité pratique, tandis que l'observation physique nécessite qu'un grand nombre d'enseignants soit physiquement présent avec les apprenants. Ces prérequis sont en contradiction avec les télé-TPs qui tendent à offrir un apprentissage à distance. La mesure immédiate de la performance dans le contexte des télé-TPs implique donc la définition d'une métrique qui s'appuie sur les actions menées par les apprenants pendant le TP.

Pour restituer cette performance, les tableaux de bord munis de multiples fonctionnalités de visualisation et d'exploration semblent être une exploitation fréquente de l'analyse de données. SAM, proposé par (Govaerts et al. 2012), FORGE (Mikroyannidis & Gomez-Goiri 2015), Realizeit (Howlin & Lynch 2014) et SocialLearn (Ferguson & Shum 2012) sont des outils proposant une ou plusieurs formes de visualisation de données d'apprentissage. Toutefois, Realizeit et LearningLocker (l'outil de visualisation utilisé dans FORGE) sont propriétaires et n'offrent pas la possibilité de les intégrer au sein d'un autre environnement comme le nôtre.

D'autre part, les outils SocialLearn ou Realizeit ne reposent pas sur des formats de données interopérables, et ne sont donc pas capables d'intégrer d'autres traces que celles pour lesquelles ils ont été conçus. En revanche, SAM ou LearningLocker reposent respectivement sur les formats Contextualized Attention Metadata (CAM) et Tin Can Api (xAPI), et peuvent ainsi traiter n'importe quelle donnée qui respecte ces formats.

L'awareness pour les situations d'apprentissage pratique nécessite une synchronisation entre les actions réalisées par les apprenants et les informations qui leur sont retournées. Ces informations doivent refléter l'instant présent, ce qui implique un traitement en pseudo temps-réel et la mise à jour automatique des visualisations. Dans SAM, les données exploitées par les tableaux de bord semblent être mises à jour de manière asynchrone (à la demande de l'utilisateur). Par contre, LearningLocker ou (Hecking et al. 2014) proposent des architectures s'appuyant sur l'utilisation des WebSocket pour retourner un feedback immédiat.

Les modèles de conception et technologies utilisées peuvent également nuire à la possibilité d'intégration.

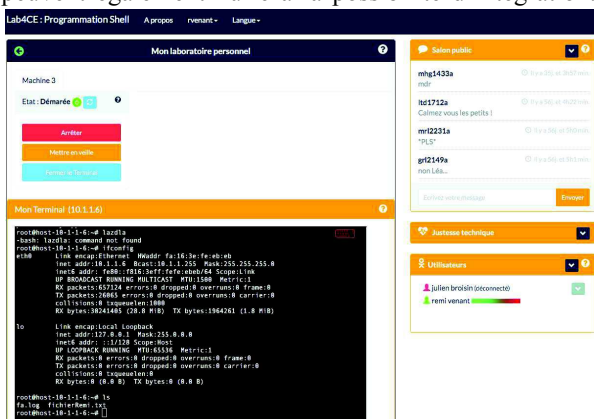


Figure 1. L'Interface Riche d'Apprentissage

Par exemple, SocialLearn repose sur un ensemble d'outils fortement couplés. La réutilisation de ces outils est donc contrainte par les autres entités, voire à une réingénierie de l'application.

Enfin, ces approches exposent les données à travers des tableaux de bord détaillés, obligeant les apprenants à alterner entre ces interfaces de visualisation et celles dédiées à l'apprentissage. Des formes plus simples de visualisation, conçues comme des composants légers tels que ceux proposés dans le projet ROLE (Santos et al. 2010), diminueraient la charge cognitive supplémentaire. En effet, (Kirsh 2000) suggère d'éviter l'affichage simultané de différents composants complexes qui amènent l'apprenant à diviser son attention pour appréhender les multiples informations.

Ces différentes visualisations ne peuvent donc pas être réutilisées sans adaptation à notre contexte. Notre objectif est donc d'identifier et de concevoir des métriques et des visualisations adaptées au contexte des activités pratiques. La section suivante introduit notre environnement de télé-TPs et analyse comment la performance des apprenants peut y être inférée.

La performance dans Lab4CE

La présentation de Lab4CE se concentre ici sur la fonctionnalité de collecte et d'analyse de données qui permet la conception, l'inférence, le stockage et la réutilisation d'indicateurs pédagogiques tel que le niveau de performance de l'apprenant.

Lab4CE : un EIAH pour l'apprentissage pratique de l'informatique

Lab4CE (Broisin et al. 2015) est un environnement de télé-TPs pour l'apprentissage de l'informatique qui propose différents services pour améliorer l'expérience utilisateur et l'engagement des apprenants lors de leurs activités pratiques. Cet EIAH s'appuie notamment sur un gestionnaire de *cloud computing* (informatique nuagique) pour la gestion des ressources, ainsi que sur une plateforme web collaborative pour faciliter leur manipulation à travers une Interface Riche d'Apprentissage (IRA). Les apprenants contrôlent leurs machines virtuelles (i.e., les démarrer, les arrêter ou les mettre en veille), créées à la volée selon le sujet du TP, et envoient des instructions à travers un terminal web proposé dans l'IRA illustrée par la figure 1. L'IRA intègre aussi des fonctionnalités de communication grâce à une messagerie instantanée et un système collaboratif pour permettre aux apprenants de travailler simultanément sur une même ressource virtuelle.

Au sein de Lab4CE, différentes informations sont collectées : les connexions à la plateforme, les messages instantanés, les invitations de collaboration entre apprenants, les actions/instructions sur les machines, et la navigation entre les différentes interfaces de l'IRA. Ces données peuvent être analysées pour étudier comment les apprenants manipulent les ressources pour réaliser le TP, et donc pour inférer une métrique de performance.

```

"object": {
  "id": "http://traffic.irit.fr/xapi/objects/computerinstruction",
  "definition": {
    "name": { "en-GB": "computer instruction", "fr-FR": "computer instruction"},
    "description": {
      "en-GB": "an instruction for a computer.",
      "fr-FR": "Une instruction d'ordinateur."
    },
    "extensions": {
      "http://exemple.irit.fr/xapi/activityextensions/computerinstruction": {
        "command": "rm", "arguments": "-v myfile"
      },
      "type": "http://exemple.irit.fr/xapi/activities/computerinstruction",
    },
  },
  "result": {
    "response": "rm: myfile: No such file or directory"
  }
}

```

Figure 2. Instruction au format xAPI

Le modèle de traces

La spécification xAPI, utilisée aujourd'hui dans de nombreux projets (Murray et al. 2012; Rabelo et al. 2015), a été adoptée pour concevoir notre modèle de traces. xAPI propose une structuration flexible des données pour représenter les actions d'utilisateurs sur des ressources : un enregistrement xAPI (*statement* en anglais) se compose d'au moins trois éléments : l'acteur (l'utilisateur), le verbe (l'action) et l'objet (la ressource). Il peut aussi comprendre la date de réalisation de l'action, son contexte ou son résultat.

Pour représenter une instruction exécutée sur une machine distante, nous avons conçu un objet xAPI spécifique. Par exemple, l'instruction « `rm -v myfile` », dont la réponse retournée par la machine sur laquelle elle a été exécutée est « `rm: myfile: No such file or directory` », est représentée par un objet *instruction* et son résultat associé, illustrés par la figure 2.

Le framework d'analyse de données

Le framework illustré par la figure 3 a pour objectif la génération et le stockage d'enregistrements xAPI, ainsi que l'enrichissement de ces derniers par l'inférence d'indicateurs. Il s'inspire d'infrastructures existantes telles que celle proposée dans le projet Migen (Santos et al. 2010), ou encore dans l'approche flexible et extensible proposée par (Hecking et al. 2014). Cependant, à la différence de ces approches, notre architecture réside essentiellement côté client afin de bénéficier du potentiel du calcul distribué sur les postes clients connectés à la plateforme.

Notre proposition inclut trois couches de composants

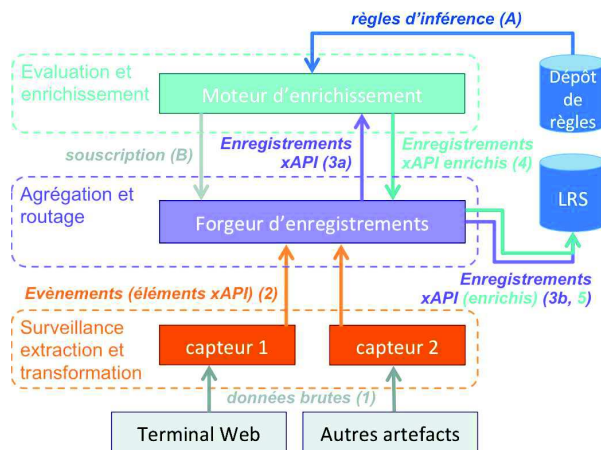


Figure 3. Le framework d'analyse de données

faiblement couplés côté client, ainsi que deux dépôts distants. Les capteurs surveillent les flux de données sur des composants spécifiques de Lab4CE ((1) sur la figure 3), génèrent des éléments xAPI et les envoient au forgeur d'enregistrements sous forme d'événements (2). La figure 3 illustre deux exemples de capteurs utilisés dans Lab4CE : le capteur 1 surveille le terminal et envoie un événement à chaque instruction exécutée. Cet événement comprend 4 éléments : le verbe (« exécuter »), l'objet (l'instruction envoyée), le résultat (la réponse retournée) et l'estampille horaire. Le capteur 2 surveille les composants de l'IRA qui gèrent les informations relatives à l'identité de l'utilisateur et au TP auquel il participe.

Le forgeur agrège les différents éléments en enregistrements xAPI, qui sont alors routés soit au moteur d'enrichissement (3a), soit directement au dépôt d'enregistrements d'apprentissage (*Learning Record Store*, LRS en anglais) (3b).

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). Ensuite, 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 composants côté client ont été développés avec AngularJS, un framework Javascript MVC (modèle-vue-contrôleur) populaire. Ce type de framework facilite la création des capteurs puisqu'il permet de lier de façon transparente le modèle de données utilisé dans les capteurs et celui de la structure DOM d'une page web, manipulée par l'utilisateur. Les deux dépôts ont été conçus sous la forme d'une pile applicative comprenant une base de données noSQL (MongoDb) pour faciliter l'intégration de nouvelles structures d'enregistrement, ainsi qu'une couche d'accès Spring Java pour exposer à la fois une interface d'accès REST et des point d'accès full-duplex (i.e., Web Socket, Ajax/XHR Streaming). Pour exploiter les traces un composant peut utiliser l'interface REST de manière asynchrone, ou souscrire à un flux d'enregistrements xAPI en temps-réel via un point d'accès full-duplex.

La justesse d'une instruction comme métrique de performance

Les données traitées par le framework contiennent notamment les instructions exécutées par les apprenants sur les ressources virtuelles, ainsi que les réponses correspondantes. Ces séquences d'instructions représentent le déroulé de l'activité pratique réalisée par l'apprenant et reflètent donc sa progression au sein du TP. Ces instructions peuvent être automatiquement évaluées comme techniquement justes ou fausses : la réponse retournée par la ressource distante fournit des informations qui peuvent être utilisées pour inférer l'état de son exécution. Dans le reste de cet article, nous nous référerons à cet état comme la *justesse technique* : une instruction qui est techniquement juste signifie qu'elle a été correctement exécutée.

Pour inférer cet indicateur, nous avons identifié les

différents patrons de messages d'erreurs qui peuvent survenir dans un terminal quand une commande techniquement fautive est exécutée. A partir de cette étude, 4 règles (R) ont été définies pour alimenter le moteur d'enrichissement de traces : R1 révèle les erreurs survenant lorsque les arguments d'une commande sont incorrects ; R2 détecte les erreurs relatives à l'appel d'une commande inconnue ; R3 et R4 s'appliquent aux erreurs spécifiques au manuel d'une commande. Finalement, une règle de plus haut niveau infère notre indicateur en combinant ces 4 règles dans un prédicat mathématique. Cet indicateur expose ainsi la justesse technique d'une commande.

Pour valider le calcul de notre indicateur, nous avons manuellement évalué la justesse technique d'un échantillon de 300 commandes produites par les étudiants. Sur la base de cet échantillon, 298 commandes furent évaluées de façon identique par le moteur, soit dans 99.3% des cas.

Outils de visualisation pour l'awareness

Les outils de visualisation présentés ici s'appuient sur la justesse technique des instructions exécutées par les apprenants, et visent à fournir une connaissance de leur performance afin de leur permettre d'analyser ce qu'ils font et pourquoi ils le font.

L'outil de comparaison social

Les outils de comparaison sociale sont conçus pour doter les apprenants d'une conscience sociale sur la performance de chacun d'entre eux. Ces outils permettent ainsi aux membres d'un groupe de comparer leur progression par rapport à celle de leurs partenaires (Michinov & Primois 2005).

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.

Cet outil illustré par la figure 4 est intégré dans

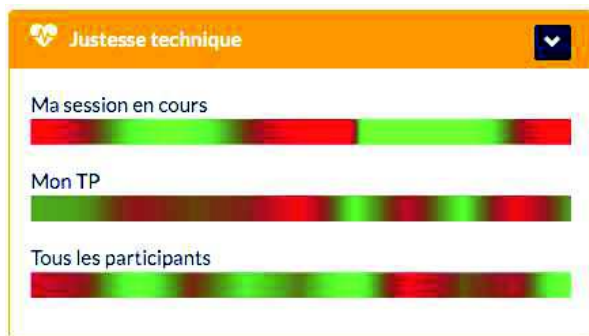


Figure 5. L'outil de comparaison sociale

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 exprime le cumul de cette performance pour l'ensemble des sessions de travail réalisées par l'apprenant sur le TP qu'il réalise ; la dernière expose ce même cumul, mais pour l'ensemble des apprenants inscrit au TP courant. Ainsi, ce mode de visualisation demeure opérationnel que les étudiants travaillent en même temps, ou de manière désynchronisée, puisque 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 ainsi que celle de leurs pairs, avec 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. Cependant, cette visualisation simple ne permet pas aux apprenants d'analyser en détail leurs propres actions ainsi que celles de leurs pairs ; l'outil suivant répond à ce besoin.

L'outil de réflexion à posteriori

Selon Boud et al. (1985), la réflexion est un processus qui consiste à repenser les expériences effectuées, les réévaluer, et apprendre de ces processus de réévaluation. Par ce modèle, les apprenants deviennent conscients de leur progression dans leur apprentissage, et capables de prendre des décisions appropriées pour l'améliorer. Ce processus peut être encouragé par les TICE à travers des outils de réflexion à posteriori (*reflection-on-action* en anglais) (Wilson & Wing 2008), définis par (Davis et al. 2005) comme l'analyse des processus d'apprentissage après que les actions aient été effectuées.

Pour revoir une session de travail, nous proposons une interface illustrée en figure 5 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



Figure 4. L'outil de réflexion à posteriori

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.

Bien que ce tableau de bord ne puisse être utilisé compétitivement avec l'IRA (l'apprenant ne peut porter son attention que sur l'une ou l'autre des interfaces), il engage les apprenants dans un processus de réflexion en soulignant les difficultés rencontrées, et permet aux tuteurs de revoir les actions des apprenants et d'évaluer leur réussite. Si l'on se réfère aux catégories d'awareness proposées par Gutwin et al. (1995), permettre à l'apprenant de savoir ce que les autres ont fait pour réaliser la tâche fait partie de l'awareness de l'espace de travail (*workspace awareness*).

Ces deux outils, implantés dans Lab4CE, ont été expérimentés dans un contexte d'apprentissage réel.

Experimentation : utilisation des outils

L'expérimentation a été réalisée à l'Université Toulouse III, et a impliqué 32 étudiants de première année inscrits dans un cours sur les commandes Linux. Les tâches à réaliser concernaient la manipulation de fichiers et de répertoires, la gestion des droits utilisateurs, et la gestion des processus. Avec Lab4CE, chaque étudiant pouvait accéder quand il le souhaitait, à sa propre machine. Les 3 sessions pratiques de 90 minutes chacune réalisées pendant cette étude se sont déroulées en présentiel. Les apprenants devaient également remettre deux comptes-rendus : un après la première session, le second en rapport avec les seconde et troisième sessions.

Le tableau 1 décrit l'utilisation des deux outils d'awareness ; les utilisateurs actifs sont ceux qui ont exécuté au moins 30 instructions dans une même session de travail. Les outils ont été utilisés par la majorité des étudiants actifs. L'outil de réflexion à posteriori a été utilisé de manière homogène pour analyser son propre travail, tandis que son utilisation pour analyser le travail réalisé par des pairs a diminué au fil des sessions. Cette diminution peut être expliquée par l'augmentation du niveau de performance des apprenants : meilleurs ils sont, moins ils ont besoin d'étudier le travail des autres. De même, l'outil de comparaison sociale, masqué par défaut dans l'IRA, a été affiché par la plupart des utilisateurs et semble suivre la même tendance d'utilisation.

Session pratique	1	2	3
Nb. d'utilisateurs actifs	17	32	30
% d'utilisateurs actifs dans le groupe	53.1	100	93.8
Outil de Comparaison Sociale (OCS)			
% d'utilisateurs actifs qui ont affiché l'OCS	76.5	59.4	60.0
Score moyen de performance sur une échelle de 0 à 100	84.8	86.0	87.6
Outils de réflexion à posteriori			
% d'utilisateurs actifs qui ont analysé leur propre travail	41.2	43.8	40.0
% d'utilisateurs actifs qui ont analysé le travail de pairs	64.7	46.9	30.0

Tableau 1. Statistiques d'usage des outils

En plus des statistiques du tableau 1, une autre information intéressante est le nombre d'analyses de travaux de pairs effectuées le jour où le premier rapport devait être rendu. Presque 43% des étudiants ont analysé au moins une session d'un autre apprenant en utilisant l'outil de réflexion, ce qui semble montrer une tendance à la recherche d'aide exécutive (i.e., obtenir directement la réponse au problème sans entrer dans un processus de réflexion (Gall 1981)). Pour limiter cette tendance, il sera nécessaire dans nos travaux futurs de contrôler l'accès à cet outil en fonction de l'apprenant.

Conclusion

Nous proposons dans cet article deux outils d'awareness qui visent à exposer aux apprenants leurs niveaux de performance pendant qu'ils réalisent une activité de travaux pratiques. Un outil d'awareness sociale révèle les niveaux courants et généraux de performance, et permet aux apprenants de se situer par rapport au groupe de participants. La technique de visualisation simple employée par cet outil le rend utilisable pendant une session pratique, alors que les utilisateurs réalisent leur activité d'apprentissage, sans requérir une attention particulière. D'autre part, l'outil de réflexion à posteriori permet aux apprenants d'analyser plus en détail leur propre travail ainsi que celui de leurs pairs.

Ces deux outils s'appuient sur un framework d'analyse de données d'apprentissage générique, exploitant le format xAPI et intégrant un moteur d'inférence capable de générer des indicateurs pédagogiques sur les données collectées. Ces outils ont été intégrés dans la plateforme Lab4CE et expérimentés dans un contexte d'apprentissage réel. La plupart des apprenants ont utilisé les deux outils, relativement à leur niveau de performance.

Toutefois, la tendance détectée à la recherche d'aide exécutive doit être adressée pour faire prévaloir une entraide constructive entre apprenants. Aussi, les instructions, évaluées uniquement selon leur justesse technique, devraient l'être également d'un point de vue sémantique en considérant le contexte pédagogique de l'activité pratique : une commande peut être techniquement juste sans pour autant être pertinente pour l'objectif attendu. Ce calcul de la justesse sémantique nécessite la production de solutions aux activités pratiques, soit manuellement par les tuteurs et enseignants, soit automatiquement. Outre cet indicateur de performance, les traces collectées dans notre système pourront être utilisées pour inférer d'autres indicateurs indépendants du domaine d'apprentissage. Les travaux de (Dimitrakopoulou 2004; Jermann & Dillenbourg 2008), qui proposent des modèles et outils pour l'analyse des interactions entre apprenants, représentent une première piste de réflexion.

Enfin, cette exploitation des différentes traces ouvre également la voie au profilage d'apprenants et à la fouille de motifs séquentiels qui représentent d'autres domaines d'investigation en concordance avec l'évaluation de la performance des apprenants.

Références

- Achumba, I. E., Azzi, D., Dunn, V. L., & Chukwudebe, G. A. (2013). Intelligent performance assessment of students' laboratory work in a virtual electronic laboratory environment. *Learning Technologies, IEEE Transactions on*, 6(2), 103-116.
- Arnold, K. E., & Pistilli, M. D. (2012, April). Course signals at Purdue: using learning analytics to increase student success. In *Proceedings of the 2nd international conference on learning analytics and knowledge* (pp. 267-270). ACM.
- Broisin, J., Venant, R., & Vidal, P. (2015). Lab4CE: a Remote Laboratory for Computer Education. *International Journal of Artificial Intelligence in Education*, 1-27.
- Boud, D.; Keogh, R. and Walker, D. (1985). What is Reflection in Learning? In *Reflection Turning Experience into Learning*, pp. 7-17.
- Cortez, J. E., Esche, S. K., Chassapis, C., Ma, J., & Nickerson, J. V. (2011). Process and learning outcomes from remotely-operated, simulated, and hands-on student laboratories. *Computers & Education*, 57(3), 2054-2067.
- Davis, D., Trevisan, M., Leiffer, P., McCormack, J., Beyerlein, S., Khan, M. J., & Brackin, P. (2013). Reflection and Metacognition in Engineering Practice. *Using Reflection and Metacognition to Improve Student Learning*, 78-103.
- De Jong, T., Linn, M. C., & Zacharia, Z. C. (2013). Physical and virtual laboratories in science and engineering education. *Science*, 340(6130), 305-308.
- Dimitrakopoulou, A. (2004). State of the art on Interaction and Collaboration Analysis (D26. 1.1). *EU Sixth Framework programme priority*, 2.
- Ferguson, R., & Shum, S. B. (2012, April). Social learning analytics: five approaches. In *Proceedings of the 2nd international conference on learning analytics and knowledge* (pp. 23-33). ACM.
- Gall & S.N.-L. (1981). Help-seeking: An understudied problem-solving skill in children. *Developmental Review*, 1(3), 224-246.
- Govaerts, S., Verbert, K., Duval, E., & Pardo, A. (2012, May). The student activity meter for awareness and self-reflection. In *CHI'12* (pp. 869-884).
- Gutierrez-Santos, S., Mavrikis, M., & Magoulas, G. (2010, June). Layered development and evaluation for intelligent support in exploratory environments: the case of microworlds. In *Intelligent Tutoring Systems* (pp. 105-114). Springer Berlin Heidelberg.
- Harward, V. J., Del Alamo, J. A., Lerman, S. R., Bailey, P. H., Carpenter, J., DeLong, K., & Long, P. D. (2008). The ilab shared architecture: A web services infrastructure to build communities of internet accessible laboratories. *Proceedings of the IEEE*, 96(6), 931-950.
- Hecking, T., Manske, S., Bollen, L., Govaerts, S., Vozniuk, A., & Hoppe, H. U. (2014). A flexible and extendable learning analytics infrastructure. In *Advances in Web-Based Learning-ICWL 2014* (pp. 123-132). Springer International Publishing.
- Howlin, C., & Lynch, D. (2014, October). Learning and Academic Analytics in the Realizeit System. In *E-Learn: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education* (Vol. 2014, No. 1, pp. 862-872).
- Jermann, P. & Dillenbourg, P. (2008). Group mirrors to support interaction regulation in collaborative problem solving. *Computers & Education*, 51(1), pp.279-296.
- Kirsh, D. (2000). A few thoughts on cognitive overload. *Intellectica*, 1(30),19-51
- Lowe, D., Newcombe, P., & Stumpers, B. (2013). Evaluation of the use of remote laboratories for secondary school science education. *Research in Science Education*, 43(3), 1197-1219.
- Michinov, N., & Primois, C. (2005). Improving productivity and creativity in online groups through social comparison process: New evidence for asynchronous electronic brainstorming. *Computers in human behavior*, 21(1), 11-28.
- Mikroyannidis, A., Gomez-Goiri, A., Domingue, J., Tranoris, C., Pareit, D., Gerwen, V. V., & Marquez-Barja, J. (2015). Deploying learning analytics for awareness and reflection in online scientific experimentation. In *5e Workshop in Awareness and Reflection in TEL (CEUR 2015)* (pp. 105-111).
- Murray, K., Berking, P., Haag, J., & Hruska, N. (2012). Mobile Learning and ADL's Experience API. *Connections: The Quarterly Journal*, 12(1), 45.
- Prensky, M. (2011). Khan Academy. *Educational Technology*, 51(5), 64.
- Rabelo, T., Lama, M., Amorim, R. R., & Vidal, J. C. (2015, October). SmartLAK: A big data architecture for supporting learning analytics services. In *Frontiers in Education Conference (FIE), IEEE* (pp. 1-5).
- Ritzhaupt, A. D., Pastore, R., & Davis, R. (2015). Effects of captions and time-compressed video on learner performance and satisfaction. *Computers in Human Behavior*, 45, 222-227.
- Salzmann, C., Govaerts, S., Halimi, W., & Gillet, D. (2015, February). The Smart Device specification for remote labs. In *Remote Engineering and Virtual Instrumentation (REV), IEEE* (pp. 199-208).
- San Cristobal, E., Castro, M., Harward, J., Baley, P., DeLong, K., & Hardison, J. (2010, April). Integration view of web labs and learning management systems. In *Education Engineering (EDUCON), 2010 IEEE* (pp. 1409-1417). IEEE.
- Wilson, J., & Jan, L. W. (2008). *Smart thinking: Developing reflection and metacognition*.