Observing a web based learning activity: a knowledge oriented approach

Amélie Cordier, Fatma Derbel, Alain Mille

Observing a web based learning activity: a knowledge oriented approach

Amélie Cordier, Fatma Derbel, Alain Mille

LIRIS UMR CNRS 5205

Université Lyon1

Université de Lyon

Introduction

Technologically Enhanced Learning (TEL) potential has been demonstrated since the very beginning of computer science history [Skinner, 1965]. Two different points of view emerged: Skinner proposed to automate the transmissive tasks of the teacher; Piaget, on the contrary, proposed harness TEL to associate learners in the learning process according to the developmental learning theory (see [Kolb, 1984] and [Rezeau, 2001] for details). These two approaches were already deeply discussed in 1964 [Oléron, 1964].

In both theories, and since the very beginning, TEL environments were used to provide learning activities selected according to some learning strategies, and taking into account the productions of the learner and, in a more general way, his/her behavior. The goal was to "automatically" guide the learner throughout the learning process. In order to provide such guidance, these environments had to incorporate observation tools for monitoring learning activities. These tools were used to assess learners' progress and to guide the learning activities accordingly. Observations collected in TEL environments have to be modeled and represented so that they can be further reused by digital systems for various tasks such as assessment, diagnosis, adaptation, rewarding, panification, etc. It should be noted that the observation of a learning process goes way beyond TEL environments. Even in situations where no TEL are involved, teachers make observations, record them and reuse them for pedagogical purposes.

Even if the observation of a learning process has always played a central part in pedagogical practices, research on that topic has not always been very active. For some years now, observation and dynamical (and sometimes real-time) exploitation of these observation is a research question of a growing importance. With the multiplication of MOOCs (Massive Online Open Courses), this research question becomes even more important and raises new challenges such as the ability to observe thousands of learners involved in the same learning activity and to build relevant knowledge and services based on these observations. Among these elements, we can list: indicators, dashboards, learner profiles, etc. Learner profiles are often built for teachers and tutors, but some systems make an effort to make their semantics clear for learners too (see for example The Observer[1]). There is a very significant literature on the subject of making easier the exploitation of this knowledge in the classroom, specifically in connection with learners profiles engineering [Brusilovsky & Millan, 2007], [Ginon et al, 2011].

In this paper, we address "observation" as a research question. More precisely, we study the concepts underlying instrumentation, collection and representation of observations in web based TEL environments. We develop models and tools according to this conceptualization. Obviously, observation and interpretation are strongly related. In the literature, it is generally admitted that "interpretation knowledge" is an abstract knowledge. Thus, the question raised is "how to connect this abstract level knowledge to low level observations?". Establishing such a connection often lead to the implementation of top-down *ad hoc* observation processes, deeply integrated in TEL environments. Top-down approaches for implementing observation mechanisms are rather complex to implement (see recommendations of the Alberta University for building profiles[2]), and some researchers, such as [Choquet & Iksal, 2007], propose to integrate specification of what has to be observed in the authoring process in order to elaborate learning indicator.

TEL systems can provide a lot of information about the learning activity, and all of them implement at least some *ad hoc* observation modules in their code and most of the recent LMS make a clear distinction between these observation modules and the others. Epiphytic learning assistants ([Ginon et al, 2014]) are developed on top of TEL environment. They rely on event-listeners that observe interactions and trigger assistance events when explicit interpretation rules are satisfied. Observation and assistance approaches are not limited to TEL environments. For instance, standard computer environments and files (operating systems, web browsers, server logs, databases, etc.) provide a lot of information that can be used by such assistants. Keyloggers can also be used for observation and assistance. For example, accessibility assistants[3] use these observation sources for providing alternative ways to interact with digital environments. In summary, many information sources are available in the learning environment, and this learning environment should not be reduced to the TEL environment itself.

As far as we know, it seems that existing TEL environments encapsulate their own observation services. As a

1 http://www.noldus.com/office/fr/observer-xt

2 http://education.alberta.ca/media/1233960/6_ch3%20learner.pdf

3 http://lib.colostate.edu/about/website-accessibility

consequence, any interpretation has to be designed and implemented within the environment. For learners as for teachers, the semantics of observations and associated indicators is specific to the environment and cannot be neither explicited nor modified easily. This black box approach, quite simple for the user, is nevertheless a problem for negotiating the semantics of what can be observed between the different actors: learners, teachers, tutors, researchers, designers, managers, etc. Even worse, it is almost impossible to build common semantics when creating groups for collective learning sessions or for peer assessment, for instance.

This can explain that, although the usage of learners traces has been studied for a while, they are rarely used to feed learner profiles for regulation, diagnosis, adaptation, assessment, and organization, in recent TEL environments. Moreover, there are only few operational TELs allowing one to provide indicators which can be adapted to the various learning situations.

Expliciting the semantics of observations in order to build relevant knowledge is a difficult process for many reasons. We have two main hypotheses. Our first hypothesis is that the source of information (e.g. learning traces) used to build indicators, dashboards and adaptation rules is not a source of "knowledge". If learning traces were represented as knowledge sources usually are (e.g. with a model) then it would be easier to explicit knowledge and processes required to perform advanced reasoning and to conduct interpretations on these traces. Our second hypothesis is that it should be mandatory that users producing interaction traces are aware of the fact that they are being observed, and, more importantly, that they should be able display, manage and share their own traces at their convenience. This property essential to ensure the proper appropriation of the environment by its users and to better understand the benefits that modeled traces bring. Consequently, we claim that the ability we have to browse our activity in a reflexive way has efficient meta-cognitive effects ([Baker & Lund, 1997]) on understanding and appropriating a learning process.

This paper studies these two hypotheses. We designed and developed an architecture and a framework for collecting observations in the form of modeled traces in web based TEL environments. Our architecture is modular and generic. We made sure to guarantee the following properties, in order to fit our claim:

1. an open collecting process, taking into account not only the TEL environment itself, but also the web environment of the learner, so as to integrate as much information as possible about the learning activity;
2. a dynamic modeling of the collected traces, in order to make it easier to explicit interpretations;
3. the ability to explain indicators and the ability to browse the corresponding interpretative process;
4. the ability to provide a reflexive process for traces collection and management allowing appropriation of his/her traces by the learner. Traces are private information and have to be understood as such.

This paper starts with a study of current work on traces in learning environments, with a special focus on MOOCs, because they are good representants of recent evolutions of teaching and learning habits. This study is built according to the four properties we identified above. Then, we present TraceMe, and we demonstrate its usage with the trace-based system called kTBS. We show why modeled traces are very important if we want to make it easier the manipulation of knowledge available in those traces, and if we want to make them usable by various users: teachers, learners, and researchers. In conclusion, we discuss the benefits of such an approach and we show why this approach renew the way we can design personalization, diagnosis, and remediation tools.

Existing observation processes

As stated above, any TEL system, and beyond that, any Learning Management System, has a specific observation module in order to collect information about the learning process. Most of them provide some analytics to their users, and most of the times, there are processes to export the collected data at the end of the session. Researchers worked on the collected data and developed methods and tools for analyzing them.

A recent report on MOOCs Data Access for research[4] demonstrated that it is, in theory, relatively easy to access the collected data. Moodle, Canvas, Open Edx, Claire, Coursera, and Claroline Connect have been considered (September 2014). All these systems have some *ad hoc* services to exploit the collected data. Most of them are able to export their data according to some models. Efforts are done to guarantee some interoperability between data issued from different platforms. For example, the IMS Global Learning Consortium provides so called Learning Tools Interoperability™[5], and the Stanford University proposes a standard for representing learning data through a generic model MoocDB [Dernoncourt et al, 2013] [Veeramachaneni et al, 2013]. This is very useful and important for designers and researchers and a recent survey is available on a lot of Learning Analytics models, methods and tools [Cooper, 2013] [Pena-Ayala, 2014] while another survey focuses on the corresponding research challenges [Ferguson, 2012]. Some other recent papers point on the fact that these Learning Analytics are not really under control of the teachers [Fletcher, 2013]. Even if the situation evolves continuously, if these data are becoming better defined, and if the processes to exploit and to export them are now more powerful and easy to use, it seems that the learner experience of his/her traces is very poor.

4 http://liris.cnrs.fr/coatcnrs/wiki/lib/exe/fetch.php?media=sous-theme_acces_donnees_v7.pdf
5 http://www.imsglobal.org/lti/ltiv1p2pd/ltiCIMv1p0pd.html#_Toc377545875

This situation of low awareness and of very low control of the learner on his/her learning traces is now recognized as an important research question and a recent document on anonymity and ethical questions reports some of these issues[6]. In a collective document, a group of international researchers pointed clearly this issue [Dillenbourg et al, 2014]. In short, it is show that including the learner in the observation process is a research issue and a key for the future if we want to ensure ethical uses of the collected data.

Moreover, if the effort is important to capitalize the existing data, there is not a lot of research on the observation step itself: how to model what is directly observed in the learner environment and how to build interactively some useful semantics all along the learning process? [Reigeluth, 2014] concludes his paper about digital traces by: “Tracking the processes through which digital traces are programed, modeled and visualized calls for a critical sociology of the structural relations that determine the conceptions and functioning of digital technology.” This remark is particularly important in the domain of TEL systems, because learning traces are necessary for managing adaptation, personalization, assessment, etc. and that the interpretation processes are never explicitly given to the actual users of a TEL system (learners, teachers, tutors).

In addition, there is an evidence of the meta-cognitive effects observed during a computer mediated learning process [Azevedo, 2005], [Khosravifar & Azevedo, 2013]. It is clear that it is important to keep and enhance this meta-cognitive effect for giving TEL systems more efficiency and to enable learners to appropriate them easily.

In [Cordier et al, 2013], it is showed that if traces can be represented as pieces of knowledge, with an explicit model, then it is possible to use a Trace Based Reasoning approach for reasoning about experience. How to get this knowledge oriented property for representing collected traces? This work demonstrates the importance of representing traces as knowledge containers of the learner experience, which is an approach that we follow in our work.

In their paper, [Elkiss et al, 2013] proposed a first approach to model traces for facilitating the computation of human learning indicators. In their framework, the modeling process is encapsulated in a one shot transformation computation for computing indicators. The structure of the raw traces is explicit, but the traces are not independent objects which can be requested as a knowledge container. E. Gendron, in [Gendron, 2010] goes one step further by proposing a method for managing the indicators in a generic framework.

[Khodabandelou, 2013] developed a sophisticated framework to manage decision making processes based on interaction logs. This tool does not make any assumption on the semantics of the logs, but proposes different heuristics for building interpretations by mining them.

[Louifi et al, 2014] proposed to extend the observation process to the learner environment, allowing to take into account what is not logged in the TEL systems. This is an interesting point, but there is no way to manage these traces as knowledge containers, and there is no reflexive way to see the traces for the learner.

[Ji et al, 2014] proposed a dynamic dashboard with an interesting approach for personalizing the way to see the learner activity. Traces are not knowledge oriented and, if the visualization process is nice, there is no way for the learner to change the way things are interpreted. There was an evidence of the interest to give the control to the learner in [Cram et al, 2008] and [Cram et al, 2007] where the learner awareness was the main goal of the approach. These approaches were based on Trace-Based Management systems which draw inspiration from the framework described in [Georgeon et al, 2011]. They did not consider directly the initial observation process as research issue, but just as a necessary thing to do in an *ad hoc* way.

In [Zaitsev, 2010], the observation step is considered explicitly through a key loggers application. Authors make a criticism of this approach because of the lake of context to interpret the real activity. The necessity of preparation of the observation process is pointed in [Wittaker et al, 2007] in order to provide an effective support for capturing and retrieving interaction information. Sellen et al stresses on the importance of the observation step in [Sellen & Wittaker, 2010] where they recommend a rational approach for life-logging.

The task tracker system [Stumpf et al, 2005] is a general framework trying to connect any learner interaction to a specific and specified taks of the learner. This is a top down modeling approach, with the teacher and designer point of view. There is no way for the learner to give his/her own interpretation, and the interactive dashboards are not open for modification by the learners.

In the following sections, we present our framework that satisfies the good properties we identified before, and lacking in most of the existing systems. These properties are: traces are designed as knowledge containers; interpretation of

6 http://liris.cnrs.fr/coatcnrs/wiki/lib/exe/fetch.php?media=fun_gt_anonymat_et_e_thique_de_la_recherche_rapport_final_10_09.pdf

traces is made explicit; learners are aware of the tracing process; learners control their learing traces; and last, the interaction tracing process is reflexive.

An open and knowledge oriented observation framework for web education

In this section, we present our contributions to answer the main issues which were pointed out above: how to represent the collected data as knowledge containers on which it is easy to elaborate different interpretations, and how to make accessible the tracing system to the learner, how to integrate the web activity with the server activity and how to give the control of his/her trace to the learner. We present the general framework which has been enhanced during this research, and a first framework to expose traces to learners.

A knowledge oriented framework for traces

The kernel for Trace Based System (kTBS) is an open-source software[7] implementing the core functionalities required to build an application tapping on interaction traces. More specifically, kTBS stores and handles four kinds of objects:

- Trace models describe the kind of information that can be expected in a given (set of) trace(s). Each trace model can be seen as a particular point of view on a kind of traced activity.
- Stored traces are traces that are fed to kTBS by external collectors. Every stored trace is related to a trace model.
- Computed traces (or transformed traces) are automatically built by kTBS, by applying a specific computation on the content of other traces, the sources of the computed traces (which can themselves be stored or computed). Every computed trace is related to a trace model, which can be different from the models of its sources (different point of view).
- Methods provide a reusable computation specification, which can be shared among several computed traces.

kTBS is built as a RESTful web service, which makes it easy to interact with from any application, regardless of their operating system or programming language. It stores its data in the RDF[8] data model, which provides the flexibility and extensibility required to account for various trace models. It can exchange information in various formats : the different concrete syntaxes of RDF, but also JSON[9], a popular format among web developers, and we are working on supporting other popular trace formats, such as Activity Streams[10] and the TinCan[11].

An open collecting process for tracing the whole learner activity: TraceMe

In this section, we describe a system for collecting modeled trace during a learning activity. This system gathers data from two collect sources: interactions on the server-side of an application and interactions on the client-side, e.g. in the browser. These interactions events are stored as “obsels” into traces managed by the kTBS described above. The main characteristic of our approach is that is collects not only events produced by the platform but also events that happen outside of the platform (e.g. events in social networks, web searches, etc.).

Architecture of the tracing system

In order to demonstrate the feasibility and the interest of modeling traces as knowledge containers and the importance of giving the full control of the tracing process to the learner, a framework (Trace-Me) has been developed and implemented. This framework has been developed by associating researchers and MOOCs designers for prototyping features and interfaces.

The architecture of the tracing system TraceMe is sketched in Figure 1. The framwork is composed of two collectors: a server side collector (Bundle_TraceMe) and the client side collector (web extension_TraceMe). The server side collector (Bundle_TraceMe) is a bundle that listens to the log events of the platform and sends them to KTBS as obsels. It also allows to generate the tracing information (URL of the trace base and trace name) from the user identity. The client side collector (TraceMe) is designed as an extension for web browsers (available for Firefox and Chrome). It allows to collect the actions of the user on the browser (click, mouse over, selection, etc.) according to a configuration. It sends these events to be recorded to the kTBS through an HTTP-POST request. A protocol of communication by cookies is implemented between these two collectors so they can send the events collected to the same activity trace.

7 http://tbs-platform.org/ktbs
8 http://www.w3.org/TR/rdf-primer/
9 by using JSON-LD http://json-ld.org/
10 http://activitystrea.ms/
11 http://tincanapi.com/

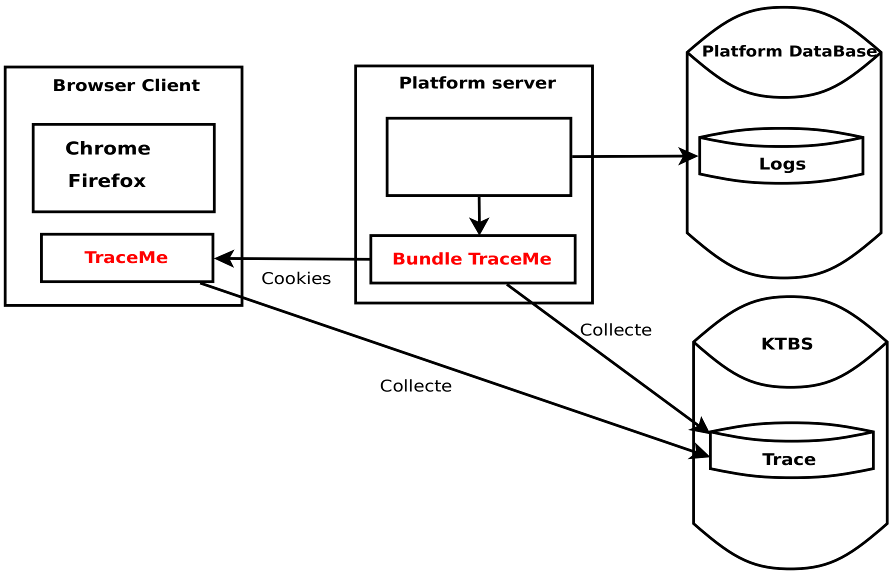

Figure 1. The architecture of the tracing system TraceMe.

Implementation of the web extension TraceMe

MOOCs are part of the web environment. The MOOC platform is the entry point of a course, but the web is largely used to achieve the course, to organize the work, to collaborate with others, to find other education resources. TraceMe is a web extension developed with the framework kangoextensions [12]. This framework allows the generation of extensions for popular browser (Chrome, Firefox, Internet Explorer and Safari) using Javascript. Figure 2 shows the architecture of the web extension TraceMe.

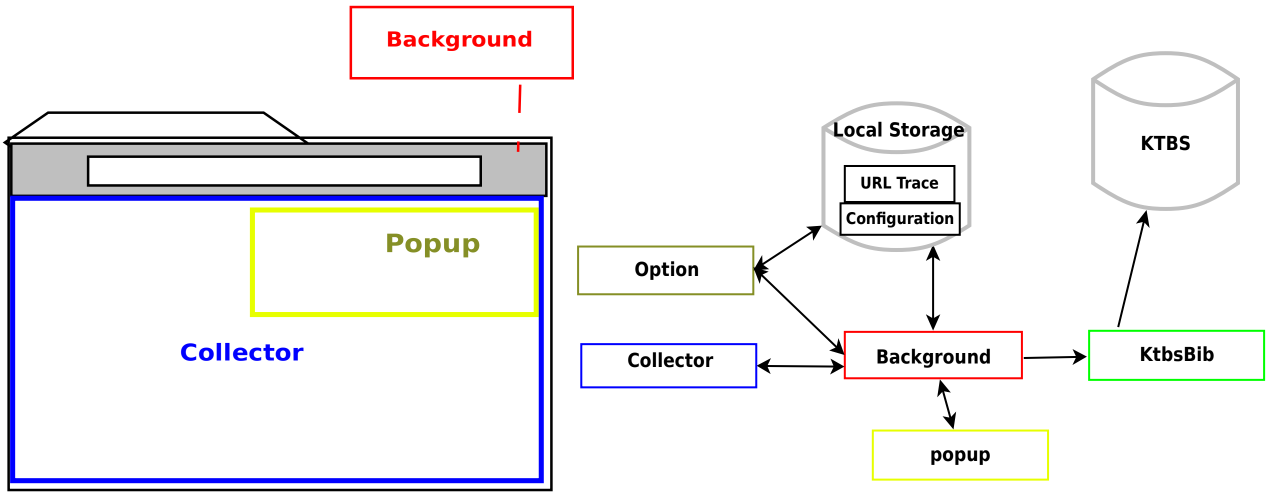

Figure 2. Architecture of the web extension TraceMe.

The tracing process has to be under the control of the learner. TraceMe uses local storage for storing trace information (URL of the trace and the configuration of the tracing). This storage method transmits tracing information to the collector by using message passing. The collector is a script that runs on any web page opened by the user. This script detects all user events stored in the configuration variable and sends them to the background which transforms them in obsels and sends obsels to ktbsBib.js which connects the kTBS.

12 http://kangoextensions.com/

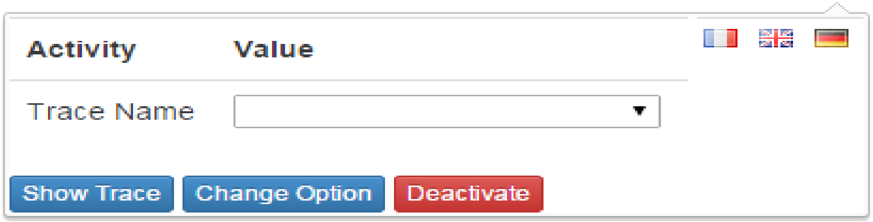

Figure 3. Popup interface.

Popup is the interface (Figure 3) that appears when the user clicks on the extension icon. This interface allows him to access the various features of TraceMe (open the options form, disable tracing, open assistant to show the activity collected, etc.).

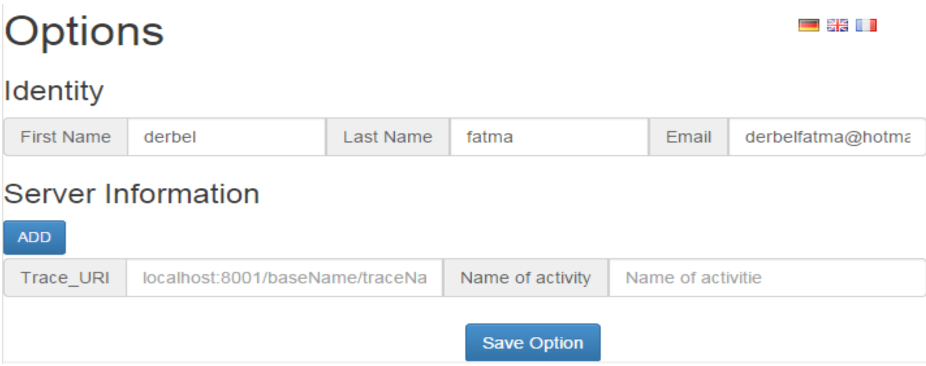

Figure 4. Options interface.

The options interface (Figure 4) is the page dedicated to user identity and server information. Users can add more traced activities, after what they should select the name of the traced activity in the extension popup. If a server collector is available, the server information will be filled automatically.

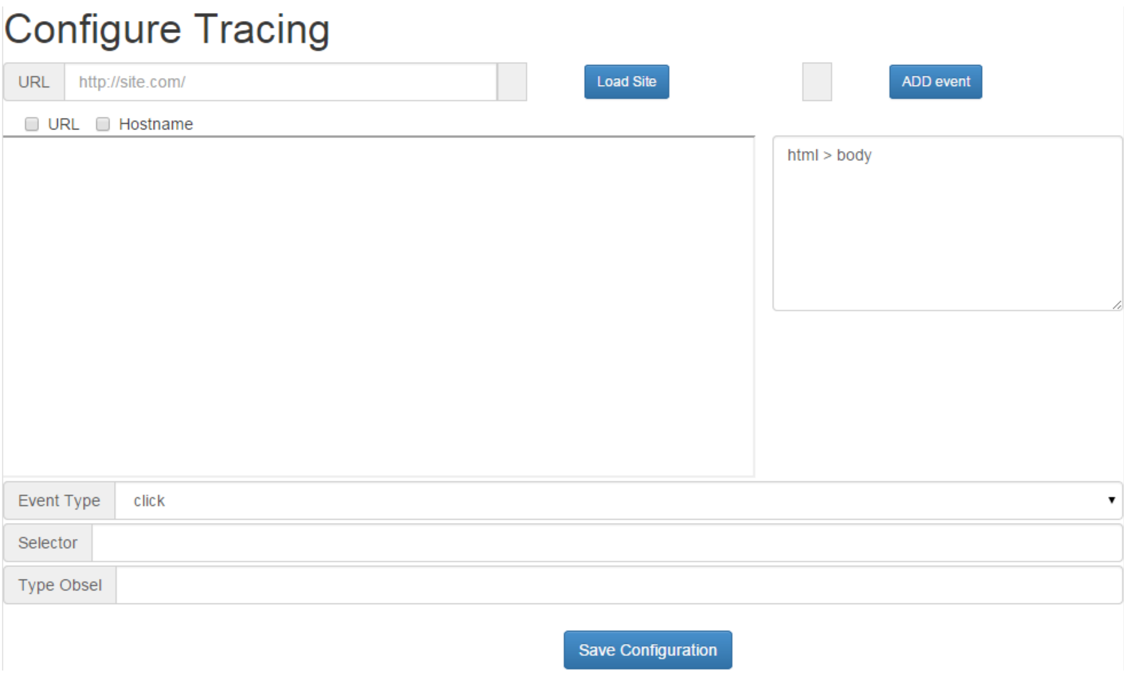

Figure 5. Configuration interface.

TraceMe provides a tracing configuration interface (Figure 5) from which one can specify the event that he/she wants to collect. The default settings is to collect only URL of visited web pages. More elements to be traced can be easily added.

Illustration of the tracing system TraceMe in the platform Claroline Connect

The learner has to be authenticated for managing his/her traces with privacy protection. An implementation of the tracing system TraceMe is performed on the learning platform Claroline Connect. Following the event connection (Figure 6) with the Registration ID on the platform Claroline Connect, the Bundle TraceMe provides the identity of the trace base (Trace Base ID = Username).

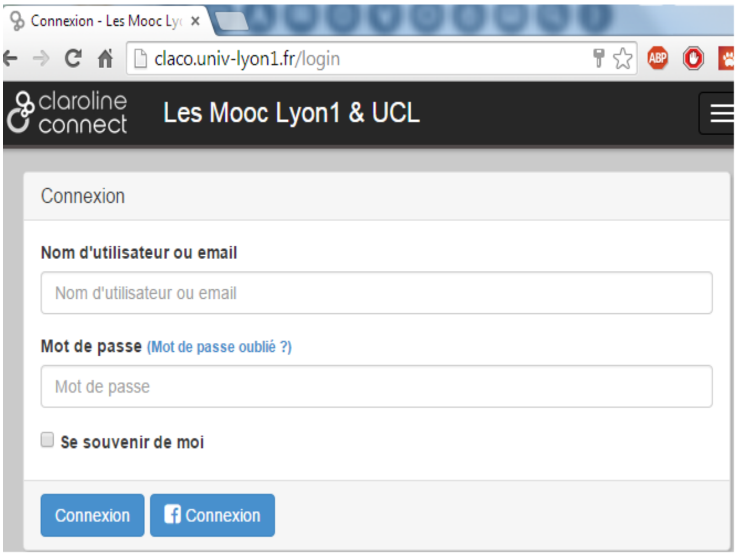

Figure 6. The connection interface of the platform Claroline Connect.

Once connected to the platform, TraceMe shows a notification for explaining that no activity is set yet.

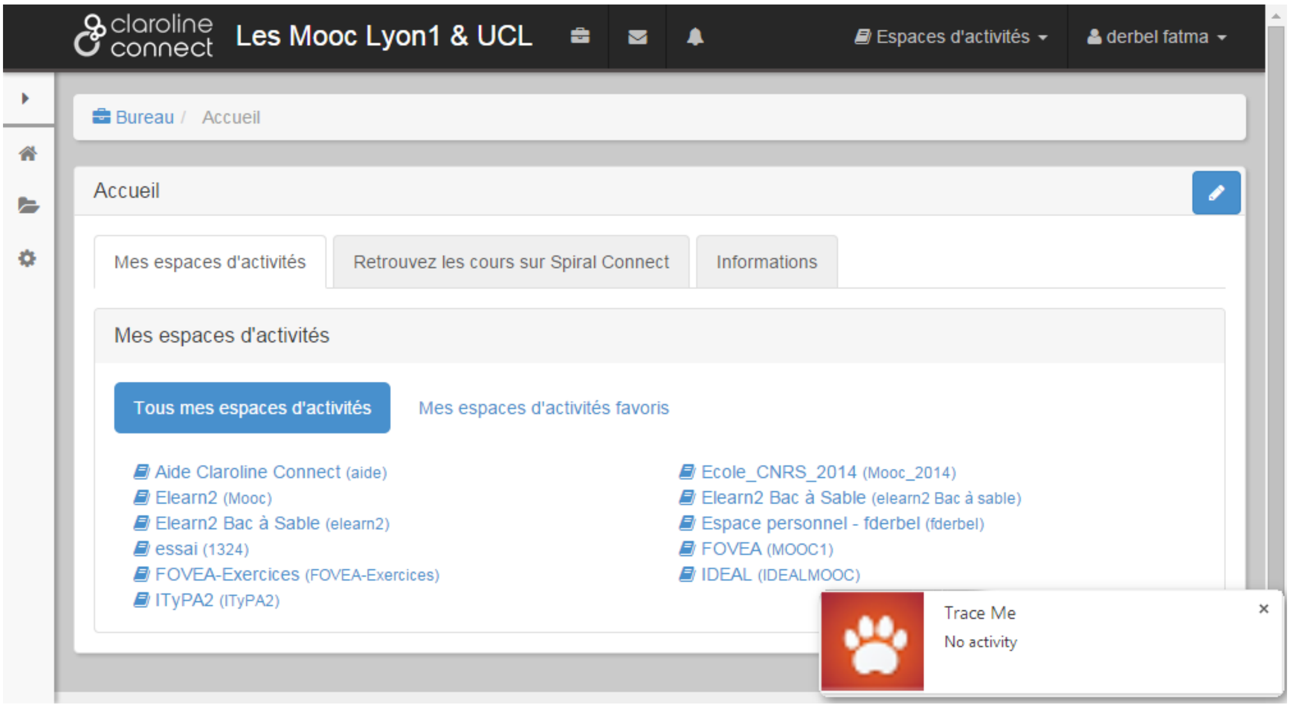

Figure 7 : The connection interface of the platform Claroline Connect

Choosing a workspace such as a MOOC (FOVEA, IDEAL, etc.) on the platform provides the identifier of the trace (trace ID = ID of the activity) to manage. The bundle TraceMe sends the ID of the trace to the TraceMe extension. It displays a notification containing the name of the activity and opens the assistant SamoTraceMe to visualize activity traces (Figure 8). This assistant has for first functionality to present his/her traces to the learner, with facilities for visualizing them according to what he/she wants to investigate in his/her learning process.

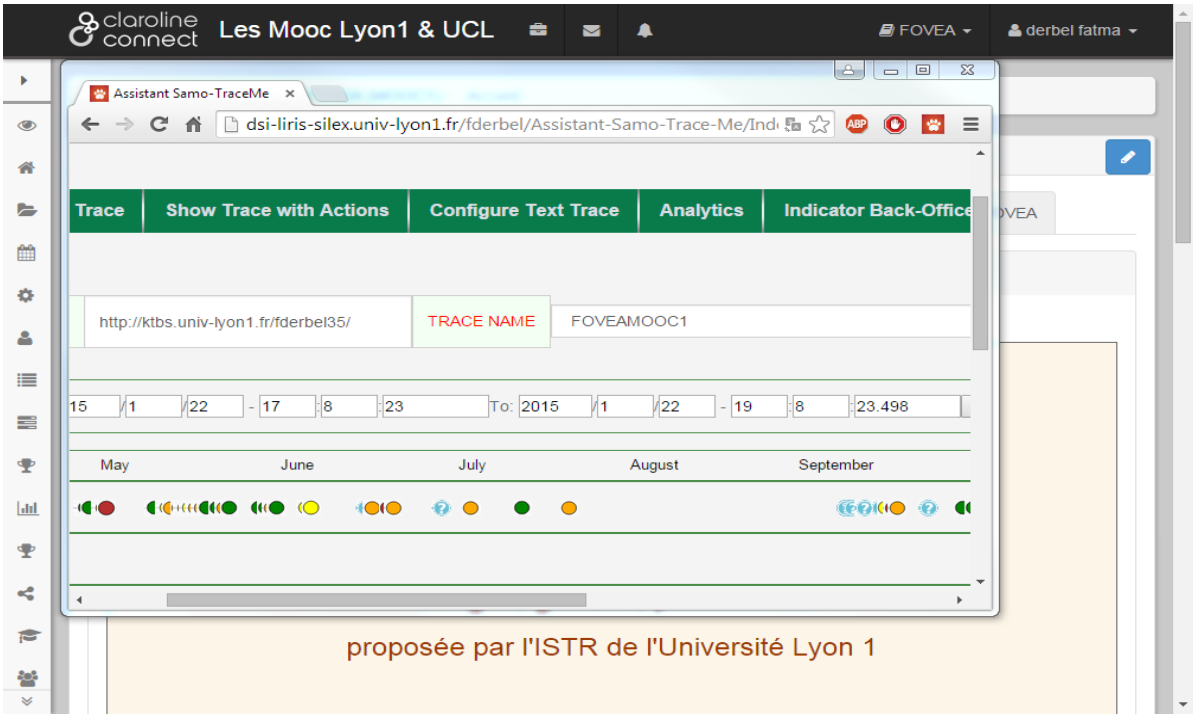

Figure 8. The assistant SamoTraceMe.

Conclusion

In this paper, we presented contributions for providing a new way to manage learners' traces. We claim that learners are first users of their traces, and any interpretation of these traces have to be explicit, shared, reusable and under the control of the learner.

The TraceMe framework is a first attempt to demonstrate that this approach is possible, and that it works at a MOOC scale. This system has been checked with the Claroline Connect platform which provides the learner authentication and which has been enriched with a TraceMe listener. The learner starts the tracing process explicitly, and his/her identity is checked for collecting his/her traces for a given activity[13] (a MOOC, a course, an activity space, etc.) offered through the Claroline Connect portal. The learner can install a TraceMe plugin in his/her browser, and when activating it explicitly, the plugin can gather his/her activity on the client side and the events logged on the server side. Two MOOCs have been used to design and to check the contributions at a significant scale (3000 learners on the first one, and 900 learners on the second).

A next step in the research agenda is to organize an experimentation with MOOCs learners not only to make a proof of concept, but to validate other hypothesis on the meta-cognitive efficiency of reflexivity of the learning process itself, and to use the framework for validating other researches about adaptation, personalization, personal assistance, assessment, certification, and so on.

This framework has been enriched with an assistance framework, for providing modeled traces oriented services: interactive visualization, learning analytics, indicators engineering.

This work opens a lot of exciting perspectives about learners' privacy management, traces semantics co-construction, good practices for interactive elicitation, indicators semantics explanation, sharing experience and practices, connecting modeled traces (local semantics) to learning oriented ontologies (global semantics), building collective traces for teachers or for collaborative learning, etc., enhancing the Trace-Base Management system performances (semantic web approach), enhancing the learner experience of his/her learning traces, and providing tools for learners, teachers, tutors, researchers for requesting M-Traces bases.

This work has been realized in the context of a CNRS mission (COAT[14]: Connaissance Ouverte à Tous) and interested readers can visit the wiki of the mission for more information, reports, access to the code, etc. This mission was supported by CNRS, University Lyon1, ENS-Lyon and Université de Lyon.

13 https://github.com/fderbel/
14http://liris.cnrs.fr/coatcnrs/wiki

References

Azevedo, R. (2005). “Computer Environments as Metacognitive Tools for Enhancing Learning,” Educ. Psychol., vol. 40, no. 4, pp. 193–197, 2005.

Baker, M. & Lund, K. (1997). “Promoting reflective interactions in a computer-supported collaborative learning environment,” in Journal of Computer Assisted Learning, 1997, vol. 13, pp. 175–193.

Brusilovsky, P. & Millán E. (2007).“User Models for Adaptive Hypermedia and Adaptive Educational Systems,” in The Adaptive Web, vol. 4321, P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds. Springer Berlin Heidelberg, 2007, pp. 3–53.

Choquet, C. and Iksal, S. (2007). “Modeling Tracks for the Model Driven Re-engineering of a TEL System,” J. Interact. Learn. Res., vol. 18, no. 2, p. 161, 2007.

Cooper, A.R. (2014). “Learning Analytics Interoperability-a survey of current literature and candidate standards.”, http://blogs.cetis.ac.uk/adam/2013/05/03/learning-analytics-interoperability/ (accessed 22th of January, 2015)

Cordier, A., Lefevre, M., Champin, P.A., Georgeon, O.L., and Mille, A. (2013). “Trace-Based Reasoning-Modeling Interaction Traces for Reasoning on Experiences.,” in FLAIRS Conference, 2013.

Cram, D., Fuchs, B., Prié, Y., and Mille, A. (2008). “An approach to User-Centric Context-Aware Assistance based on Interaction Traces,” LIRIS UMR 5205 CNRS/INSA de Lyon/Université Claude Bernard Lyon 1/Université Lumière Lyon 2/Ecole Centrale de Lyon, RR-LIRIS-2008-009, 2008.

Cram, D., Jouvin, D., and Mille, A. (2007). “Visualizing Interaction Traces to improve Reflexivity in Synchronous Collaborative e-Learning Activities,” in 6th European Conference on e-Learning, 2007, pp. 147–158.

Dernoncourt, F., Taylor, C., Veeramachaneni, K., and Reilly, U.O. (2013). “MOOCdb□: Developing Standards and Systems for MOOC Data Science,” 2013.

Dillenbourg, P., Fox, A., Kirchner, C., Mitchell, J., and Wirsing, M. (2014). “Massive open online courses: current state and perspectives,” Dagstuhl Manif., 2014.

Dillenbourg, P., Fox, A., Kirchner, C., Mitchell, J., and Wirsing, M. (2014). “Massively Open Online Courses, Current State and Perspectives (Dagstuhl Perspectives Workshop 14112),” Dagstuhl Rep., vol. 4, no. 3, pp. 47–61, 2014.

Elkiss, H., Khoukhi, F., and Bekkhoucha, (2013). “On modeling traces in a computing environment for human learning based indicators,” J. Theor. Appl. Inf. Technol., vol. 56, no. 1, 2013.

Ferguson, R. (2012). “Learning analytics: drivers, developments and challenges,” Int. J. Technol. Enhanc. Learn., vol. 4, no. 5/6, pp. 304–317, 2012.

Fletcher, A. (2013). “How Big Data Is Taking Teachers Out of the Lecturing Business,” Sci. Am., pp. 1–4, 2013.

Gendron, E. (2010). “Cadre conceptuel pour l’élaboration d’indicateurs de collaboration à partir des traces d’activité,” Claude Bernard Lyon1, Lyon, 2010.

Georgeon, O.L., Mille, A., Bellet, T., Mathern, B., and F. Ritter, F.E. (2011). “Supporting activity modelling from activity traces,” Expert Syst., p. no–no, Mar. 2011.

Ginon B., Champin, P.A, Jean-Daubias, S., and Lefevre, M. (2014). “aLDEAS□: un langage de définition de systèmes d’assistance épiphyte,” in Actes des 25èmes Journées francophones d’Ingénierie des Connaissances, Clermont-Ferrand, France, 2014, pp. 137–148.

Ginon, B., Jean-Daubias, S., and Lefevre, M. (2011).“Evolutive learners profiles,” in World Conference on Educational Multimedia, Hypermedia and Telecommunications, 2011, vol. 2011, pp. 3311–3320.

Ji, M., Michel, C., Lavoué, E. and George, S. (2014). “DDART, a Dynamic Dashboard for Collection, Analysis and Visualization of Activity and Reporting Traces,” presented at the 9th European Conference on Technology Enhanced Learning, Graaz, Austria, 2014, pp. 440–445.

Khodabandelou, G., Hug, C., Deneckère, R., Salinesi, C., Bajec, M., Kornyshova, E., and Jankovic, M. (2013). “COTS PRODUCTS TO TRACE METHOD ENACTMENT: REVIEW AND SELECTION,” in Proc of ECIS 2013, 2013.

Khosravifar, B. and Azevedo, R. (2013). “Adaptive Multi-Agent Architecture to Track Students’ Self-Regulated Learning,” in AIED 2013 Workshops \ldots, 2013, pp. 49–52.

Kolb, D.A. (1984). “Experiential learning: experience as the source of learning and development.” Englewood Cliffs, NJ: Prentice Hall., 1984.

Louifi, A., Bousbia, N., Azouaou, F., and Merzoug, F. (2014).“ESITrace: A User Side Trace and Annotation Collection Tool.,” New Horiz. Web Based Learn., p. 49, Jan. 2014.

Oléron, P. (1964). “Introduction à l’enseignement programmé.,” Enfance. Tome 17 N°1, pp. 1–38, 1964.

Peña-Ayala, A (2014). “Review: Educational data mining: A survey and a data mining-based analysis of recent works,” Expert Syst. Appl., vol. 41, no. Part 1, pp. 1432–1462, Mar. 2014.

Reigeluth, T. (2014). “Why data is not enough: Digital traces as control of self and self-control.,” Surveillance & Society 12(2), pp. 243–354, 2014.

Rezeau, J (2001). “Médiatisation et médiation pédagogique dans un environnement multimédia,” Doctorat Études anglaises – Didactique, Langue de spécialité, Systèmes d’Information et de Communication, Université Bordeaux 2, Bordeaux, 2001.

Sellen, A. and Whittaker, S. (2010). “Beyond total capture: a constructive critique of lifelogging,” Commun. ACM, 2010.

Skinner, B.F. (1965). “Review lecture: The technology of teaching,” Proc. R. Soc. Lond. B Biol. Sci., pp. 427–443, 1965.

Stumpf, S., Bao, X., Dragunov, A., Dietterich.T.G., Herlocker, J.G., Johnsrude, Li, L. and Shen, J. (2005). “Predicting user tasks: I know what you’re doing,” in 20th National Conference on Artificial Intelligence (AAAI-05), Workshop on Human Comprehensible Machine Learning, 2005.

Veeramachaneni, K., Dernoncourt, F., Taylor, C., Pardos, Z. and O’Reilly, U.-M. (2013). “Moocdb: Developing data standards for mooc data science,” in AIED 2013 Workshops Proceedings Volume, 2013, p. 17.

Whittaker, S., Tucker, S., Swampillai, K., and Laban, R. (2007). “Design and evaluation of systems to support interaction capture and retrieval,” Pers. Ubiquitous Comput., vol. 12, no. 3, pp. 197–221, Mar. 2007.

Zaitsev, O. (2010). “Skeleton keys: the purpose and applications of keyloggers,” Netw. Secur., vol. 2010, no. 10, pp. 12–17, 2010.