



# Trace Based System in TEL Systems: Theory and Practice

Tarek Djouad<sup>1(✉)</sup> and Alain Mille<sup>2</sup>

<sup>1</sup> Icosy Laboratory, Khenchela University, Khenchela, Algeria  
tarek.djouad@gmail.com

<sup>2</sup> Liris, Lyon1 University, UMR CNRS 5205, Lyon, France  
alain.mille@univ-lyon1.fr

**Abstract.** We present in this paper an easier way to manage activity traces and to compute human learning indicators activities in Technology Enhanced Learning TEL systems. We review our research work related to Trace based system TBS and we explain how we use TBS to develop new and generic model to represent the indicator life cycle from its creation to its reuse. This paper presents the underlying theory and how this theory is implemented to compute human learning indicators activities available for use with any other learning platform, provided the TBS can access the learning platform traces.

**Keywords:** Activity traces · Trace based system  
Human learning indicator activity · Indicators engineering  
Technology Enhanced Learning systems

## 1 Introduction

Most Technology Enhanced Learning (TEL) systems use activity traces to make diagnoses during the learning process in order to propose appropriate assistance for learners, tutors or teachers. Some TEL systems can be used by designers or teachers to set up these diagnoses and describe the information to be collected at the design step [1], but in most cases, both the data collection and diagnostic processes are embedded into the code and the IT developers are needed for designing and modifying them. In practice, therefore, it is very difficult to personalize the tracing and diagnostic processes, and, consequently, in most cases distance education systems offer very little in the way of personalization and remediation services during learning activities. In the field of online education research, researchers and analysts are also very interested in activity traces and are increasingly equipping themselves with tools to build interesting data from application logs, and then treat them as activity traces before carrying out treatments to understand current learning processes [2–4]. Learning analytics is now a big issue for researchers [5] and it is very important for the efficiency of the research to be able to get clear information on the semantics of the logs, which

remains currently a big challenge. In order to overcome these difficulties, we have been working for several years to consider activity traces as knowledge objects available to designers, teachers, tutors, learners, researchers and, in general, to those involved in learning processes. Since activity traces are constructed to analyze, diagnose and understand expected or feared learning phenomena, we have also developed representations of learning indicators and their computation processes, based on traces of modelled activities. These new implementation versions of Trace based system we propose in this paper are intended to compute human learning indicators activities. In this paper, we present the state of our current research and its latest developments, illustrated with practical examples while recalling the theoretical foundations of the modelled activity trace. We describe on the one hand the formalization of the approach and on the other hand the developed and experienced software environments. We conclude by showing the strong potential of this approach for professional, individual and opportunistic learning activities, in particular with regard to the ethical aspects of e-learning.

## 2 Activity Traces in TEL Systems

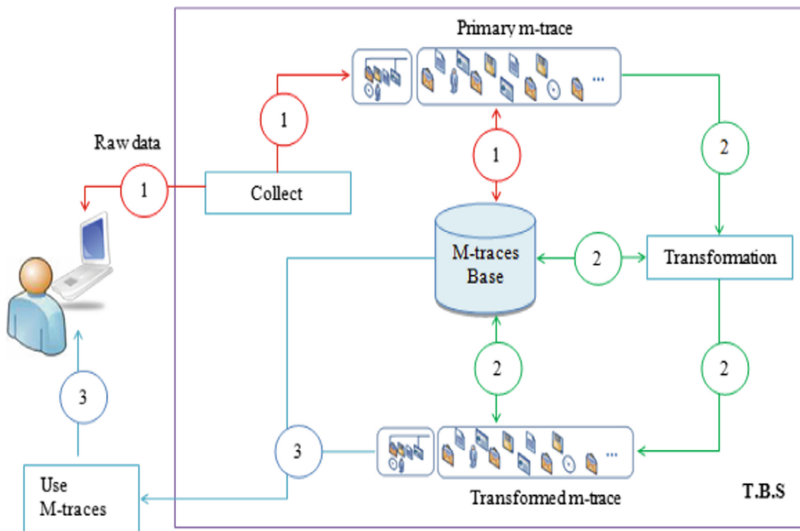
Most of learning platforms produce activity traces. These traces are saved in different formats: Logs files, Databases, video and audio files, etc. Several works propose to use and to exploit these traces in TEL systems. For example, Betbeder [6], Guraud [7] and Dyke [8] define experiments to collect multimodal data in order to analyze traces. Mazza [9], France [10] and Cram [11] visualize traces in real time and provide a feedback to teachers and learners on their own activities. Ferraris [12] and Voisin [13] propose model driven engineering approaches to understand learning scenarios using trace. As activity traces represent very important knowledge containers, it is meaningful to provide specific environment for managing them as such. Several frameworks have been developed in order to manage activity traces [14, 15] as for helping the designers to integrate them in the target application [16] or for helping researchers and analysts to understand the learning process in a batch way [17, 18]. These approaches are very useful for managing and requesting traces as data, but do not consider traces as knowledge containers per se. In this paper, we will focus on one knowledge based approach for collecting, representing, managing, requesting and transforming activity traces. As far as we know, this approach is original and we will illustrate in which extent this approach allows new usages in the context of Technology Enhanced Learning systems.

## 3 Trace Based System

The Trace based system or TBS is proposed and implemented [19] by the TWEAK [20] research group to manage modelled activity traces or m-traces [21–23]. An m-trace in TBS is a structured object: the trace model and the corresponding trace instance in the form of a sequence of observed elements or obsels. Each instance obsel part of an m-trace is temporally situated by a

time stamp and satisfies the trace model part of the m-trace. TBS proposes explicit transformation operators to be applied to a set of m-traces (transformation sources) in order to obtain other transformed m-traces (transformation targets). All m-trace obsels are represented by structured information resulting from a transformation operation using source m-trace obsels. Each m-trace is the result of some transformation of a lower level m-trace, except for the lowest level, directly built from an observation process constructing the primary m-trace. TBS proposes three steps for using m-traces (Fig. 1):

1. Users, as teachers/tutors or learners, use learning platforms. These platforms provide raw data as a source of observation. TBS connects to learning platforms, collects raw data and uses these data to build a primary m-trace (model and instance). This primary m-trace is then saved in an m-trace base,
2. TBS uses this primary m-trace and transforms it into other transformed m-traces according to the semantics of these transformations. The transformed m-trace is saved in the m-trace base. In turn, these transformed m-traces can be transformed again into other transformed m-traces. Starting from one primary m-trace, a transformation graph is progressively built and saved for providing explicit explanations of any transformation, i.e. providing the semantics of any m-trace in the m-trace base,
3. Moreover, the m-trace base can be used by any assistant to manage indicators, allow indicator computation, provide smart visualization, etc.



**Fig. 1.** Trace based system used by our research team to manage m-traces.

## 4 Case Studies: Indicator Computations Using TBS

We propose in this section some case studies of how to use TBS in real learning situations to compute human learning indicators activities. The new developed systems we propose here are based on TBS and their goal is to compute indicators activities using modelled traces. We will illustrate two systems we built: Trace based indicator management system TB-IMS, and multi-agents indicator computation system IC-MAS. But first we will illustrate the latest implemented KTBS: a kernel trace based system which is a concrete implementation of TBS.

### 4.1 Short Presentation of the Used Kernel Trace Based System (KTBS)

Kernel trace based system [24] is an implementation of TBS. KTBS considers an activity trace as a model and a set of obsels (m-trace). It allows to create m-traces bases, m-traces' models, m-traces' obsels, transformed m-traces, etc. Each obsel has a set of attributes like type, value, time-begin, time-end, etc. KTBS stores modelled traces as RDF ontology and manipulates them using a set of operators like: filter, fusion, SPARQL query, etc. It proposes to use JSON [25], REST [26] and TURTLE [27] to describe m-traces. Figure 2 presents an example of m-trace model in KTBS using Turtle syntax.

```
@prefix : <http://iris.cnrs.fr/silex/2009/ktbs#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.

<.> :contains <model1> .

<model1> a :TraceModel ;
  :hasUnit :millisecond .

<#EnterChatRoom> a :ObselType .
<#SendMsg> a :ObselType .
<#MsgReceived> a :ObselType .
<#LeaveRoom> a :ObselType .
```

Fig. 2. Example of creating a model in KTBS using Turtle syntax.

### 4.2 Trace Based Indicator Management System TB-IMS

In order to demonstrate concretely how to compute indicators activities using TBS, we developed an effective method and its implementation [28] to carry out the entire life cycle of indicators activities. We recall that in TEL systems an indicator is a mathematical variable with a set of values [29]. In our work,

we propose an indicator as composed of a model descriptor and a set of instances, and we propose four steps to describe and compute an indicator instance using a TB-IMS. Each indicator instance is computed using these four steps:

- **Step1:** To compute a new indicator, we propose to associate it with a set of empty m-traces. These m-traces will be instanced later for evaluating the indicator formula to compute indicator instances,
- **Step2:** We associate a transformation sequence to the empty indicator. The transformation sequence related to these m-traces describes how to move from a primary m-trace (described by its model) to a specific indicator (described by its formula),
- **Step3:** Collects data from the learning environment and builds the primary m-trace instance,
- **Step4:** Finally, we execute the transformation sequence, from the primary m-trace instantiated in step 3 to the indicator defined in step 1.

Indicator computation and m-trace transformation are managed by the prototype we developed: Trace based-indicator management system TB-IMS [30]. Figure 3 shows the TB-IMS collector module, the transformation module, the equation editor and the visualization result. In this example we compute the proportion between chat messages and private messages related to ‘user15’ according to a time interval T. Computation is possible without coding in machine. The system saves in its databases the indicator, its transformation, intermediate m-traces and the primary m-trace, which allows the indicator to be reused to compute new other indicators.

### 4.3 A Multi-agent System to Compute Human Learning Indicators Activities

We propose also to use Multi-agents technology to compute indicators activities based on TBS [31]. The indicator computation- multi agents system IC-MAS [32] we propose allows creating, updating, deleting, reusing and sharing indicator. These agents are important in order to:

- **Minimize the indicator’s computation complexity.** If the indicator computation is difficult (Example: the case of a high level indicator defined by sub-indicators) then our system reuses existing indicators to compute new one. We use information like transformation sequence, computation rule and category to perform this computation,
- **The need to compute a same indicator in different contexts is very high.** Several users attempt to compute the same indicator with closest time intervals. The system creates for each new computation a list of agents that will minimize the indicator computation time.

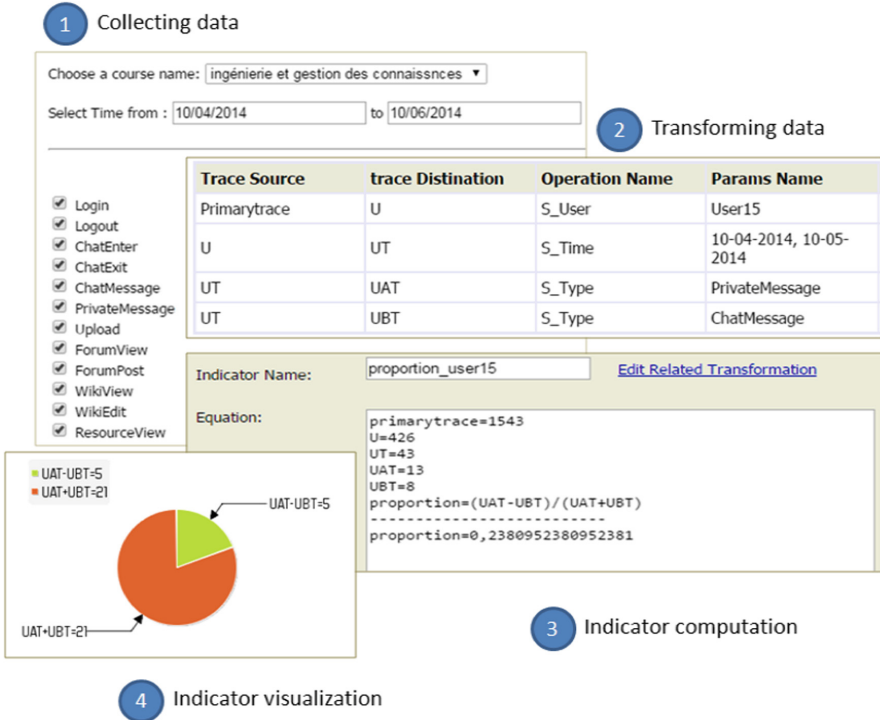


Fig. 3. Computing the proportion indicator using TB-IMS.

We propose these following agents to build our system:

- **The collecting agent** uses raw data issued from learning platforms (Example: database, Log files, etc.) to select useful data used to compute indicator. The collecting agent is guided by variables which are listed in the computation rule to optimize the collecting step. Once completed, it stores the collecting results in an m-traces base as a primary m-trace and keeps the collecting history to reuse it later. This agent aims to prepare things for having a better performance in a future collecting step,
- **The transformation agent** transforms a primary m-trace created by the collecting agent. Each transformed trace becomes a variable labelled according to the name of the transformed trace and takes its value from the number of instances present in the transformed m-trace. This agent uses transformation operators such as filtering, merging, matching, etc. to transform m-traces,
- **The computation agent** executes the computation rule using variables created by the transformation agent and saves it in the indicator base,

- **The interface agent** displays indicators using graphical views (Pie charts, Bar graphs, Histograms, etc.). It also retrieve information provided by teachers to define indicators,
- **The coordinator agent** coordinates and synchronizes the different agents to provide a consistent functioning of the system.

Figure 4 explains an example of indicator computation using IC-MAS.

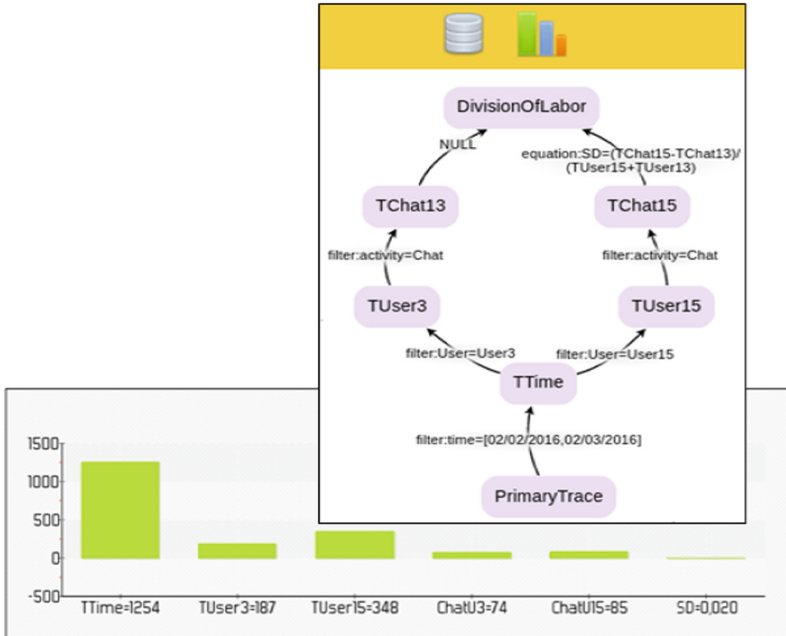


Fig. 4. Computing the proportion indicator using IC-MAS.

## 5 Conclusion

Our research was an opportunity to work on the consolidation of collaborative work around the theory of modelled traces. Indeed, we were able to experiment concretely with its potential to formalize and represent the notion of indicator as a structured computer object that provides the knowledge necessary for its elaboration. This knowledge structure integrates the modelled source trace as described in the learning environment, the set of logical and temporal transformations necessary to allow the calculation of indicators whose semantics are thus made completely explicit. This explicit representation is proposed in the form of an operational IT structure, adaptable by designers or teachers to new situations without requiring reprogramming, but requiring only a modification

of the description of the mobilized knowledge. For researchers, the availability of a basic semantics for activity traces makes it possible to integrate this knowledge into the parameterization of Learning Analytics tools, but also to design specific and documented experiments to serve the object of their research. Finally, and these are the latest developments under way in our research, the ability to explain activity traces opens up very promising solutions in terms of ethics of use of personal data by involving learners and teachers in controlling the use of activity traces during learning. Learners and tutors can even, during the activity itself, suggest new ways of interpreting their learning behavior to make easier their task or to build new knowledge about human learning processes.

## References

1. Ouali, M.A., Iksal, S., Laforcade, P.: The strategic organization of the observation in a TEL system: studies and first formalizations. In: Proceedings of the 6th International Conference on Computer Supported Education, Barcelona, Spain, pp. 317–324 (2014)
2. Georgeon, O.L., Mille, A., Bellet, T., Mathern, B., Ritter, F.E.: Supporting activity modelling from activity traces. *Expert Syst.* **29**, 261–275 (2012)
3. Ji, M.: Exploiting activity traces and learners' reports to support self-regulation in project based learning. Ph.D. thesis in Computer Science, INSA, Lyon, France (2015)
4. Lebis, A., Lefevre, M., Luengo, V., Guin, N.: Approche narrative des processus d'analyses de traces d'apprentissage: un framework ontologique pour la capitalisation. In: Proceedings of the 8ième conférence sur les Environnements Informatiques pour l'Apprentissage Humain, Strasbourg, France, pp. 101–112 (2017)
5. Learning Analytics and Knowledge Conference 2017. <http://educ-lak17.educ.sfu.ca/>
6. Betbeder, M.L., Reffay, C., Channier, T.: Environnement audiographique synchrone: recueil et transcription pour l'analyse des interactions multimodales. In: Proceedings of the premières journées communication et apprentissage instrumentés en réseau, Amiens, France, pp. 406–420 (2006)
7. Guéraud, V., Michelet, S., Adam, J.M.: Suivi de classe à distance: propositions génériques et expérimentation en électricité. In: Proceedings of the 3ième conférence sur les Environnements Informatiques pour l'Apprentissage Humain, Lausanne, Switzerland, pp. 167–178 (2007)
8. Dyke, G., Lund, K., Girardot, J.J.: Tatiana, un environnement d'aide à l'analyse de traces d'interactions humaines. *Technique et Science Informatiques.* **29**, 1179–1205 (2010)
9. Mazza, R., Botturi, L.: Monitoring an online course with the GISMO tool: a case study. *J. Interact. Learn. Res.* **18**, 251–265 (2007)
10. France, L., Heraud, J.M., Marty, J.C., Carron, T.: Visualisation et régulation de l'activité des apprenants dans un EIAH tracé. In: Proceeding of the 3ième conférence sur les Environnements Informatiques pour l'Apprentissage Humain, Lausanne, Switzerland, pp. 179–184 (2007)
11. Cram, D., Mathern, B., Mille, A.: A complete chronicle discovery approach: application to activity analysis. *Expert Syst.* **29**, 321–346 (2012)



12. Ferraris, C., Vignollet, A.L., David, J.: Modélisation de scénarios d'apprentissage collaboratif pour la classe. In: Proceedings of the 2ième conférence sur les Environnements Informatiques pour l'Apprentissage Humain, Montpellier, France, pp. 285–296 (2005)
13. Voisin, J., Vidal, P.: Une approche conduite par les modèles pour le traçage des activités. *Sciences et Technologies de l'Information et de la Communication pour l'Éducation et la Formation* **14**, 1–18 (2007)
14. Santos, O.C., Rodríguez, A., Gaudio, E., Boticario, J.G.: Helping the tutor to manage a collaborative task in a web-based learning environment. *Artif. Intell. Educ.* **4**, 153–162 (2003)
15. Avouris, N., Fiotakis, G., Kahrimanis, G., Margaritis, M., Komis, V.: Beyond logging of fingertip actions: analysis of collaborative learning using multiple sources of data. *J. Interact. Learn. Res.* **18**(2), 231–250 (2007)
16. Bouhineau, D., Luengo, V., Mandran, N.: Concevoir, Produire, Dècrire, Évaluer et Partager des Données, Opérateurs, Processus d'analyse et Résultats d'études sur l'apprentissage Humain avec Ordinateur. In: Proceedings of the 8ième conférence sur les Environnements Informatiques pour l'Apprentissage Humain, Strasbourg, France, pp. 125–136 (2017)
17. Dyke, G.: A model for managing and capitalising on the analyses of traces of activity in collaborative interaction. Doctorate thesis, Ecole Nationale Supérieure des Mines, Saint-Etienne, France (2009)
18. Toussaint, B.-M., Luengo, V., Jambon, F.: Proposition d'un Framework de Traitement de Traces pour l'Analyse de Connaissances Perceptivo-Gestuelles. (Le cas de la chirurgie orthopédique percutanée). In: Proceeding of the 7th Conférence sur les Environnements Informatiques pour l'Apprentissage Humain, Agadir, Morocco, pp. 222–233 (2015)
19. Kernel of Trace Based System. <https://kernel-for-trace-based-systems.readthedocs.io/en/latest/>
20. TWEAK Research Team. <https://liris.cnrs.fr/equipres/?id=75>
21. Settouti, L.S., Marty, J.C., Mille, A., Prié, Y.: A trace-based system for technology-enhanced learning systems personalisation. In: Proceedings of the 9th International Conference on Advanced Learning Technologies, Riga, Latvia, Italy, pp. 93–97 (2009)
22. Zarka, R., Champin, P.A., Cordier, A., Egyed-Zsigmond, E., Lamontagne, L., Mille, A.: TStore: a trace-base management system using finite-state transducer approach for trace transformation. In: Proceeding of the 1st International Conference on Model-Driven Engineering and Software Development, Barcelona, Spain, pp. 117–122 (2013)
23. Cordier, A., Lefevre, M., Champin, P.A., Georgeon, O., Mille, A.: Trace-based reasoning-modeling interaction traces for reasoning on experiences. In: Proceedings of the 26th International Florida Artificial Intelligence Research Society Conference, Pete Beach, Florida, USA, pp. 363–368 (2013)
24. KTBS. <http://tbs-platform.org/tbs/doku.php/tools:ktbs>
25. JSON. <https://www.json.org/json-fr.html>
26. REST. <https://www.w3.org/2001/sw/wiki/REST>
27. TURTLE. <https://www.w3.org/TR/turtle>
28. Djouad, T., Mille, A.: Observing and understanding an on-line learning activity: a model-based approach for activity indicator engineering. *Technol. Knowl. Learn.* (2017). <https://doi.org/10.1007/s10758-017-9337-9>

29. Dimitracopoulou, A., Petrou, A., Martinez, A., Marcos, J.A., Kollias, V., Jermann, P., Harrer, A., Dimitriadis, Y., Bollen, L.: State of the art on interaction analysis for metacognitive support and diagnosis. Technical report, Deliverable D31.1.1 Kaleidoscope Project (2006). <https://telearn.archives-ouvertes.fr/hal-00190146>
30. Trace Based Indicator Management System. [www.github.com/tb-ims/tb-ims/](http://www.github.com/tb-ims/tb-ims/)
31. Djouad, T., Mille, A.: A multi-agents system to compute human learning indicators activities based on model-driven engineering approach. *Int. J. Technol. Enhanc. Learn.* **10**(1/2), 91–110 (2018). <https://doi.org/10.1504/IJTEL.2018.10008599>
32. Indicator Computation-Multi Agent System. <https://github.com/Activity-Traces/IC-MAS>