Object-Based Trace Model for Automatic Indicator Computation in the Human Learning Environments

Bernabé Batchakui(✉), Thomas Djotio, Ibrahim Moukouop, Alex Ndouna

National Advanced School of Engineering, Yaoundé, Cameroon

`bbatchakui@gmail.com`

Abstract—This paper proposes a traces model in the form of an object or class model (in the UML sense) which allows the automatic calculation of indicators of various kinds and independently of the computer environment for human learning (CEHL). The model is based on the establishment of a trace-based system that encompasses all the logic of traces collecting and indicators calculation. It is implemented in the form of a trace database. It is an important contribution in the field of the exploitation of the traces of apprenticeship in a CEHL because it provides a general formalism for modeling the traces and allowing the calculation of several indicators at the same time. Also, with the inclusion of calculated indicators as potential learning traces, our model provides a formalism for classifying the various indicators in the form of inheritance relationships, which promotes the reuse of indicators already calculated. Economically, the model can allow organizations with different learning platforms to invest only in one trace Management System. At the social level, it can allow a better sharing of trace databases between the various research institutions in the field of CEHL.

Keywords—indicators calculation, CEHL, learning traces, trace model, trace-based system, reuse of indicators

1 Introduction

A digital trace of interaction is a temporally located series of observations [1] [4] which is either an interaction between humans, mediated in various ways by computer, or a series of actions and reactions between a human and a computer. There are several types of traces (computer data, screenshots, audio, videos). They are categorized according to the tools that are used for capturing and collecting the observed interactions. In a computer environment for human learning (CEHL), there are three facets of use of traces of interactions, namely: empirical studies on the characteristics of human interactions, reengineering of CEHLs and adaptation of interactions in real time. With the progress of work on traces of interactions, we are even witnessing a sort of theorizing of traces which are presented by certain authors as fully-fledged Computer objects. The appropriate modeling of interaction traces allows teachers to design and automate the calculation of indicators, which in turn will help them under-

stand, assess, monitor and support ongoing learning. The main contributions related the traces model are the following:

— SBT-IM, the works of Tarek Djouad [6], proposes a system for calculating indicators in Moodle. It is based on the architecture SBT (System Base Traces) [13]. The purpose of this system is to collect and generate instances of the modeled trace. It takes into account the specificities of Moodle in its implementation. It also proposes a set of operators for the transformation of the traces, and from the transformed traces it calculates the value of the indicator.

— EM-AGIIR is a multi-agent system proposed by Diagne [5] for the reuse of indicators from a new traces source. The indicator to be reused is made up of metadata and a function which presents the necessary prerequisites for the indicator's reuse.

— The indicator patterns: the work of Gendron [2], proposes a model for calculating and reusing indicators. The idea in his works is to observe user activity through digital traces of use and then to propose collaboration indicators calculated from these traces. The management of the indicators is done using a rule-based system.

— CAPTEN-MANTA work by Lebis [10], proposes meta-models to overcome the technical constraints linked to traces and their analysis processes in order to improve the capitalization of the latter.

— xapi [14][16] is the standard of the best known within the Data Learning community. It integrates the logic that emerged via SCORM [15] in relation to the notion of learner actions during a learning situation. It is a technical specification that allows recording the most varied learning experiences. Its main goal is to provide a framework for services allowing the recording, storage and exchange of user traces.

— IMS Caliper [13] proposes a model for sharing learning traces which rather integrates the logic which emerges from IMS-LD [14] in relation to the notion of actions of learners in a learning situation. It also offers a minimal format for a trace in the form of a triplet: "Actor", which, as in the case of xAPI, represents the entity performing the action; "Action", which is used to define the action carried out; and "Activity", referring to what the action focused on. One of the advantages of this standard is also to provide a semantic layer on the definition of the information contained in the trace.

The contributions above do not offer complete engineering indicators from traces from the observation of activity of learning, including: collecting evidence, modeling the track regardless of its origin, a preparation of data for the calculation of indicators, the calculation of indicators and the possibility of reusing them. The standard xapi and IMS Caliper provide models for the trace attributes to ignore the CEHL, adding a semantic layer to the information contained in the trace. But these works do not specifically concern the field of indicator calculation. They are rather oriented towards the storage of traces from several CEHL in a common database. Tarek Djouad in [7] proposed a SBT (Traces Based System) for the calculation of indicators from trace models. However, his interest was only on activity indicators. Diagne [5] and Gendron [2] deal with this indicator reuse problematic but in a reuse perspective of them in a new indicator computing platform and also the work does not cover the entire

engineering process to be implemented. The analyses of trace models presented above lead to the following central research question: which model of traces for the calculation of indicators would take into account the calculation of several types of indicators at once, independently of a learning platform, while integrating the notion of transformation and reuse of traces? The models in the standard training traces of xAPI and XMS-Caliper make it possible to have a basic trace model giving access to all the information necessary for the calculation of an indicator. A model based on traces of the objects, which comes from xapi and XMS-Caliper, can meet the following needs: Collection of data processing traces, model and reuse of the indicator. Such a model is obtained by enrichment based on the design methodology of a trace-based system (SBT) [17].

The rest of this article is made up of: a methodology section, which presents the model we propose, an experimentation section, which focuses on the experimental framework to validate the model, a result section, which presents the results obtained, a discussion section, which compares our proposal to the existing one, and a conclusion, which presents a synthesis of this work, followed by the interest for the research and some prospects.

2 Methodology

This section presents our proposed model for the automatic calculation of the following indicators: activities, cognitive, social and emotional. We rely on the process of calculating indicators from a trace model to build our trace model. We first present a plot model for each type of indicator, then we deduce a more generic model which takes into account all these different models. The various models proposed here are intended to provide all the data necessary for the application of the various calculation formulas for each type of indicator. These formulas are the basis for calculating our indicators.

2.1 A trace model for calculating activity indicators title

An activity indicator makes it possible, among other things, to have the following information for a resource and for a learner: access time, working time, density of actions, etc. The trace model represented in Figure 1 presents the class model for the calculation of indicators of a learner's activities.
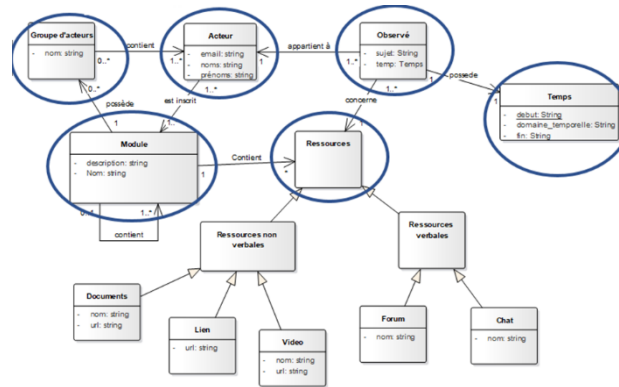
Fig. 1. Trace model for calculating activity indicator

This model is made up of the following basic classes:

— **Actor**: used to save information on the learners to whom the collected traces belong.
— **Observe**: used to represent the collected traces.
— **Time**: makes it possible to define the temporal domain of the trace, that is, the period during which the trace was collected. It is characterized by a beginning and an end, and a time unit that can be minutes, hours, days, years. In the framework of our work we take as time unit "date and time" for reasons of precision in the collection of our traces.
— **Resources**: allows one to save data on the traced resources. This table represents all the elements liable to undergo user action on the CEHL. These resources can be of two kinds: verbal resources that affect all resources that support the learner to communicate with his peers, often the case with forums or chats; non-verbal resources which concern resources which do not involve a communication of the learner, like documents, videos, links, etc.
— **Module**: This class represents the entities to which the traced resources are attached. Here it can be a CEHL, a teaching unit or a lesson in a teaching unit.
— **Group of actors**: is a class representing the different groups to which the users and traces belong, in particular when collecting traces during the performance of a collective activity.

We present here a trace model for the calculation of indicators of activity, used in the following as a basic model to determine the trace models for other indicators like the cognitive indicators.

2.2 Trace model for the calculation of cognitive indicators

Cognitive indicators provide information about the acquisition of knowledge in a learning activity. The model that we propose emphasizes the evaluation of the acquisition of knowledge by the learner. We first focus on the indicator of activity models

while refining some classes by adding certain attributes that help us achieve our goals. The classes that have been tweaked here are "Module" and "Ressources_non_verbal" (non-verbal resources). These classes have inherited additional attributes from the activity indicator model, which are «objectif_a_atteindre" (objective_to_be_achieved). For class "Module" and for the class "Ressources_non_verbales" (Non-verbal resources) we have "connaissance_a_utiliser"(knowledge to be used), "exercice"(exercise), "niveau_difficulte" (level_of_difficulty) and "reponse_utilisateur" (user_response).

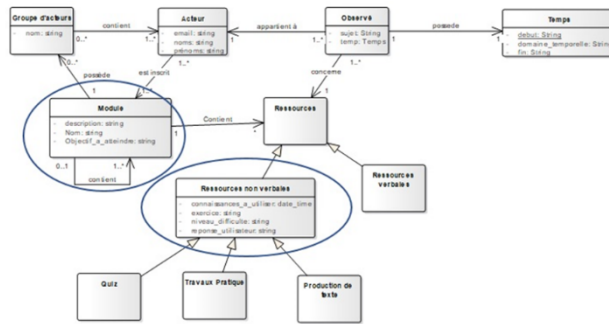Figure 2 shows the traces model for calculating cognitive indicators.

Fig. 2. Trace model for calculation of cognitive indicators

The main changes compared to the activity indicators calculation model are as follows:

— The addition of new classes that inherit the class from non-verbal resources. These new classes represent the resources that allow us to assess the level of knowledge acquisition by the user. These are in particular the resources which allow the learner to be assessed, namely: quizzes, practical work and exercises requiring argumentation on the part of the learner that we call here text production exercises.

— To know that this is the knowledge acquired by the learner, we need to know for each assessment method: the knowledge to be used in order to be able to pass the assessment successfully, the level of difficulty of the assessment, the answers expected during the evaluation and the answers given by the learner during his evaluation. So, we modified the "Ressources_Non_verbales" (non-verbal resources) by adding attributes cited for storing this information. In the event that a set of combined evaluations makes it possible to acquire precise knowledge, we added, at the level of the "Module" class, the module regrouping these different resources, which we call here "objectif_a_atteindre" (objective to be achieved).

2.3 Track model to calculate emotional indicators

Emotional indicators allow a glimpse of the emotional state of a learner or members of a group for the realization of a learning activity. Within the framework of this work we limit ourselves to the detection of the motivation of the learner during the

follow-up of a learning activity. The motivation of the learner is assessed here based on the time spent by the learner in solving a problem depending on the type of activity and the level of difficulty of the activity. Our trace model for the calculation of indicators focuses on the collection of these type of information. Figure 3 presents our trace model proposal for the calculation of affective indicators. This model is also based on the trace model for the calculation of activity indicators. We modify it by adding attributes on some classes that allow us to determine the emotional state of a learner. The added attributes are: "niveau_difficulte" (level of difficulty), "status", "temps_prevu" (time allow) and "type_activite" (type of activity).
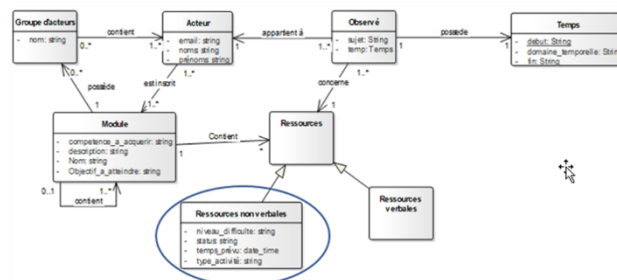

Fig. 3. Trace model for calculating affective indicator

To be able to measure the motivation state of a learner, it suffices to specify for each traced resource: the type of activity (compulsory or not) represented by the attribute "type_activite"(type of attribute), the minimum time foreseen for the realization of the activity, represented by the "temps_prevu"(estimated time) attribute, the status (activity validated or not) represented by the "status" attribute. An activity is valid when the learner has completed it. These different attributes are integrated at the level of the "Ressources_non_verbales" (non-verbal resources) class. The latter includes resources for acquiring knowledge and resources for assessing the acquisition of that knowledge.

2.4 Trace model for calculating social indicators

Social indicators allow having information on the collaboration, coordination, organization and the relationship of learning with peers. These indicators are mainly calculated either by grouping traces of activities from user groups and performing calculations for the targeted indicators, or by grouping rather indicators of activities already calculated by individual group functions of group learners. To calculate these social indicators, we will use the same trace model for activity indicators.

After having presented a basic model for the calculation of each type of indicator, this section presents a model, based on the models presented above, which allows to automatically calculate all the type of indicators presented above. We have therefore added to our model, some complementary classes, namely: Transformation Execution, DerivedMethod and Indicator. Figure 4 presents the trace model for the automatic calculation of the indicators, the complementary elements are circled.
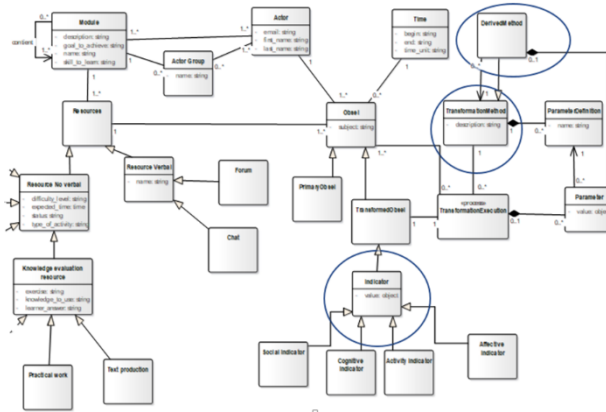
Fig. 4. Trace model for the automatic calculation of indicators

— The "TransformationExecution" class represents the execution of a transformation method on a trace or an indicator. At the end, it provides either another trace that we will call transformed trace or another indicator. It has predefined transformation methods and also gives the possibility for an appropriate user interface to be able to add other transformation methods according to the user's need. As a predefined transformation method we have:

  • *Filter*, which allows one to sort the observations, while keeping only those which satisfy a certain condition. For example, the user can decide to consider only traces included in a precise time interval for these calculations.
  • *Fusion*, which allows to make the union of several observations.
  • *Enrichment*, which makes it possible to enrich an observation by adding other attributes, other relationships or new observations.
  • *Rewriting of patterns*, which consists in creating new observations on the basis of a pattern sought from an observed source.
  • *Composites*, which consists in the composition of sequential or parallel to methods existing transformations.

— The "DerivedMethod" class represents the class of derived methods. These methods make it possible to facilitate the use of more generic methods such as those mentioned above by highlighting a particular way of using them. A derived method comprises a parent method and values for the parameters. For example, for the function for calculating the mood indicator, there is a derived method which will have as its parent method, an enrichment method, and as parameters, the algorithm for calculating the mood state and the degree of certainty.

— The "Indicator" class represents the calculated indicators. Often considered as distinct from the traces, we have decided in our framework to consider them as observed because certain indicators are reused, either, in conjunction with other indicators, or, with other observed traces, to calculate other indicators. The indicators will thus be dated by the observed data used for their calculation.

2.5 Process or algorithm for automatically calculating indicators

The algorithm (or process) for automatically calculating indicators from our model is as follows:

— For each type of indicator, the user will define a period of observation or collection of evidence necessary for calculating the target indicator.
— After defining the observation periods, a process will be responsible for verifying the end of each observation period for each indicator and performing the transformation method corresponding to the calculation of this indicator.
— The value of the indicator will be stored in the trace base for viewing by the user with the aim of helping him in his decision-making, or the stored value will be used for the calculation of another indicator.

2.6 Preprocessing module

In order to make our model independent of the source CEHL, we set up a preprocessing module for traces coming from the CEHL to disregard them, based on the concept of variable defined in [10]. The variable according to [10] is an object in the UML sense which makes it possible to extract the concepts contained in the traces instead of focusing on the values they contain. Each concept is thus modeled by a name, which makes it possible to symbolize them to the developer, and by its type (date, number, string of characters). We have added a description field to add a semantic layer. So, our preprocessing consists, first, in implementing several predefined variables in our system, and then for each trace file imported into our system, if this file is for example in Excel form, the user will have to do for each column of his file, a correspondence with one of the variables of our system. These correspondences therefore allow us to transform all the traces coming from various sources into traces directly addable in our system.

3 Experimentation and results

3.1 Installation

The practical implementation of our model is done in several steps:

— Transform our model into a database. The latter is then used as a storage medium for our traces.
— For each new source of traces, set up or implements a connector which will be linked to our database of traces. This connector will allow one to add the traces coming from the new source in our database of traces. Traces from different sources will sometimes require preprocessing to make them match our model and be stored in the database.
— Implement transformation methods for the different types of indicators to be considered.

— Setting up a display module that will be connected to our footsteps database will have display indicators calculated in an understandable form for the user.

Figure 5 shows the architecture on the general deployment of a trace-based system that will use our model as a base for the calculation of indicators. It mainly consists of 3 modules:

— The module for collecting traces from CEHL which mainly contains our connectors and our ontology mentioned above. This module is responsible for collecting the different traces and formatting them in a format that can be stored directly in our trace database.
— The trace transformation and automatic indicator calculation module which mainly contains the traces database built from our model, and a period launch system for automatic indicator calculation. The calculated launch period will depend on the type of indicators for better decision support.
— The visualization module which offers the user an adequate visualization interface of the various indicators calculated for better decision making.
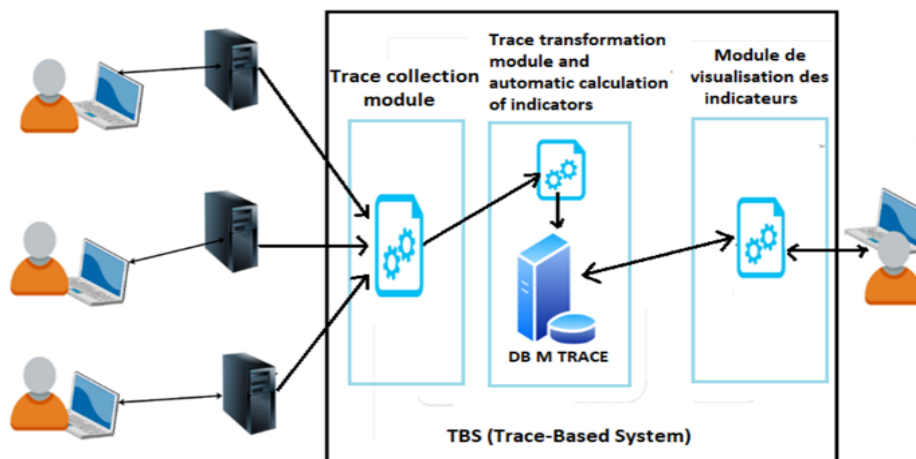
Fig. 5. Model deployment architecture

3.2 Experimental setting

To test our solution, we set up our trace-based system in a JavaScript environment using the Node JS Framework. For the visualization of our indicators we set up a web platform allowing a given user to have access to indicators in an appropriate form. We took as a source of test data the Moodle log files collected during a programming learning session in which the 3rd year computer engineering students of the National Advanced School of Engineering of Yaoundé participated during the academic year 2016-2017.

3.3 Results

This part presents some results of visualization of traces and calculation of indicators obtained after experimenting with our model.

Figure 6 shows a screenshot of the visualization of the learning traces stored in our database from our test platform. One can see the traces of interactions of certain users; it presents, the date and time when the action took place, the user who carried out the action, and a description of the action carried out.

Date et heure	Utilisateur	Action
31 December 2017 23:14	Jeatsa Armel	The user with id '12' viewed the submission with id '10' for the workshop with course module id '18'
31 December 2017 23:14	Jeatsa Armel	The user with id '12' viewed the workshop activity with course module id '18'
31 December 2017 23:08	Simon Pierre	The user with id '16' viewed the course with id '12'
31 December 2017 23:08	Simon Pierre	The user id '16' has logged in
31 December 2017 22:58	Jeatsa Armel	The user with id '12' viewed the submission with id '10' for the workshop with course module id '18'
31 December 2017 22:58	Jeatsa Armel	The user with id '12' submission with id '6' for the workshop with course module id '18'
31 December 2017 22:47	Jeatsa Armel	The user id '12' viewed the workshop activity with course module id '18'
31 December 2017 22:46	Jeatsa Armel	The user id '12' viewed the workshop activity with course module id '18'
31 December 2017 22:36	Jeatsa Armel	The user with id '12' viewed the submission with id '14' for the workshop with course module id '18'
31 December 2017 22:35	Jeatsa Armel	The user with id '12' viewed the submission with id '16' for the workshop with course module id '18'

Fig. 6. trace visualization interface

Figure 7 presents the "identification of interactions" indicator which is an indicator of activities presenting all the actions made by a given user over a specified period. In our case it represents the activity of a learner over a period of one month by presenting the number of actions performed per day and on which particular module the learner's action was performed.
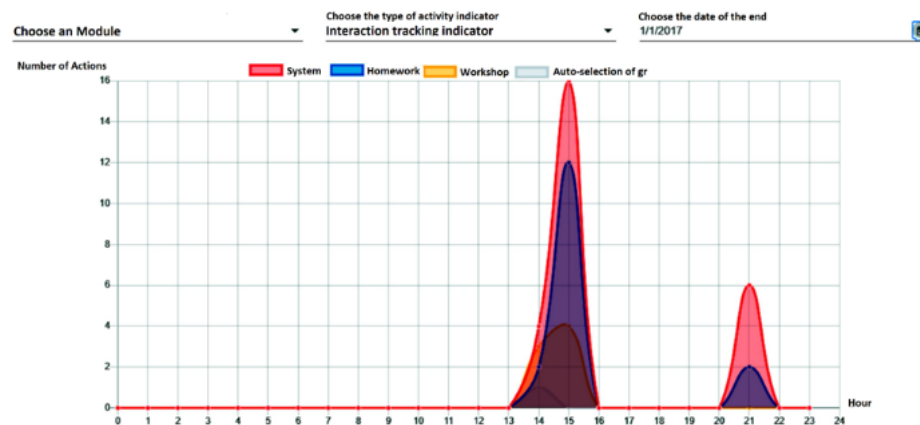
Fig. 7. graph of indicators of a learner's activities over a period of the year

Figure 8 shows the calculation result of the activity indicator "Active Agent" on a module or course "Duty" of a learning session on calculating 1 day. The indicator presented here as a curve gives information on the number of active users on this module at some point in January 04, 2017.
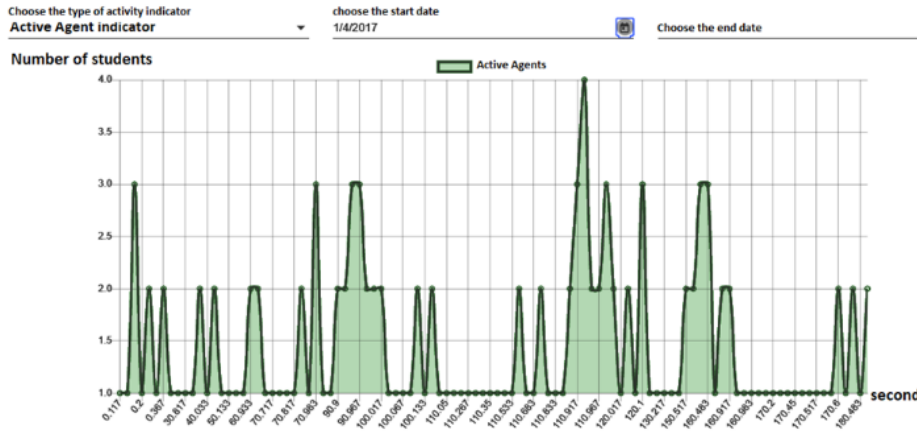
Fig. 8. Active Agent indicators on a module

Figure 9 still represents the "identification of interactions" indicator also presented in Figure 7, but here, it is calculated for a precise module. Figure 9 then shows the number of interactions made on a module (in our case the "Homework" module) over a period of 1 month.

Fig. 9. Interaction tracking indicator on a module

Figure 10 represents a cognitive indicator that we have put in place to obtain the marks obtained by the learners during the "report writing" activity. This indicator allows, for example, depending on the level where the student's grade is located, to decide wheth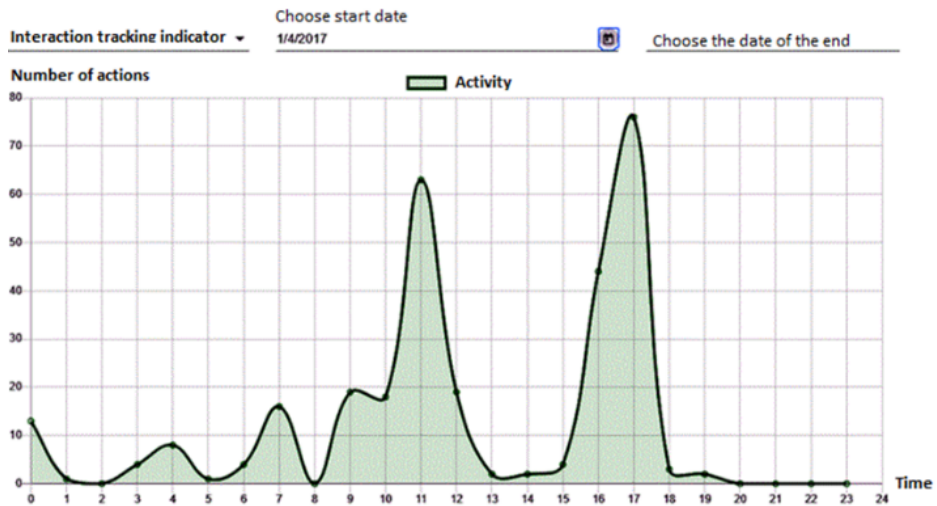er the student was able to acquire the skill that the learners were supposed to have obtained after following this module. In our case, the skill to be acquired is "Writing a software engineering report". We can for example conclude from this module that a learner whose grade is between 16 and 20 has acquired the skill. In figure 10, these learners are those whose grades are shown using a green bars. But on the other hand, we can conclude that those with marks between 0 and 5 have not acquired this knowledge. In the figure, the learners in this case are those whose marks are represented by red bars.

Fig. 10. Skill acquisition test indicator

Figure 11 represents the result of the calculation of a social or collaboration indicator, in particular the "degree of participation" indicator which allows having, for each activity where the work must be done in a group, the degree of participation of each learner in their group based on the number of interactions performed on that activity. In this case, for each group detected, it is thus possible to have the most active and least active learners. If the final grade of the learner should depend on the percentage of contribution in the group, this indicator already provides us with decision support in this case.
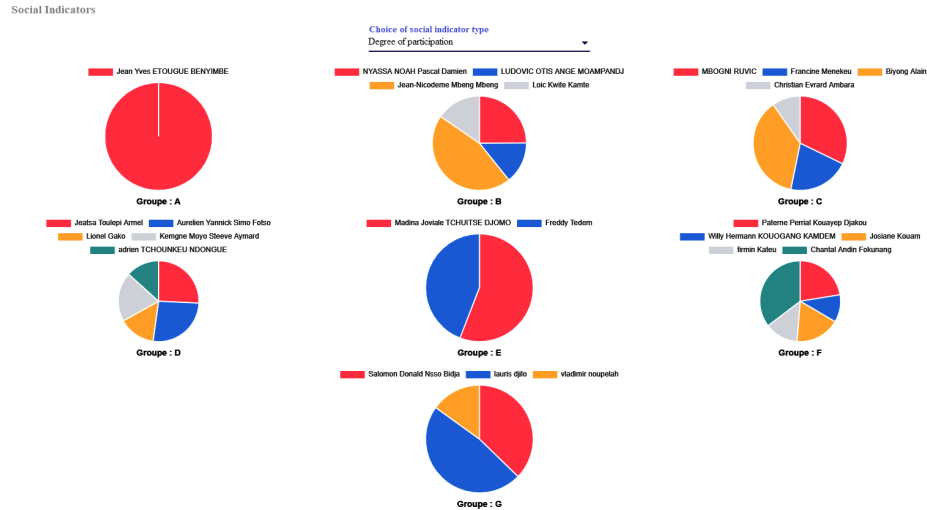
Fig. 11. Indicator of the degree of participation of members of a group in carrying out a task

4 Discussions

Table 1 presents a comparison between our model and those proposed by Diagne (2009), Gendron (2010), Lebis (2019) and Tarek (2011).

Table 1. Comparison of existing work on the modeling of traces in order to calculate indicators with our method with our method

		Gendron, 2010	Diagne, 2009	Tarek, 2011	Lebis, 2019	Our Model
Collection of data	Using a trace template	No	No	Yes	Yes	Yes
	Trace model independent of the CEHL source of the trace	No	No	No	Yes	Yes
	Using traces from other tools	Yes: SBT Modeled	No	Yes: import of traces	No	Yes
	Data explicitly collected	No	No	Yes	No	Yes
	Data implicitly collected to calculate the indicator	Yes	Yes	No	No	Yes
Trace transformation	Trace transformation	No	No	Yes	Yes	Yes
	Transformation of implicit data	Yes	Yes	No	No	Yes
	Explicit data transformation	No	No	Yes	Yes	Yes
	Device for reusing trace transformations	No	No	Yes	Yes	Yes
Modeling of the indicator	Indicator modeled by a formula	Yes	Yes	Yes	No	Yes
	Indicator modeled by a formula and trace transformations	No	No	Yes	No	Yes
	Technology used	Expert system	Multi-money	SBT	Ontology	SBT

				system			
Visualization of the indicator and traces	Real-time trace visualization	No	No	Yes	No	Yes
	Visualization of traces for the development of indicators	No	No	Yes	No	Yes
	Visualization of indicators	Yes	No	Yes	No	
Reuse of the indicator	Generation of indicators from their description (formulas and transformations)	Yes	Yes	Yes	No	Yes
	Device for reusing indicator formulas	Partially	Partially	Yes	No	Yes
	Classification of indicators	No	No	No	No	Yes
	Reuse of indicators as potential traces	No	No	No	No	Yes
	Reuse the value of a previously calculated indicator for the calculation of another	No	No	No	No	Yes
	Import-export of traces and associated knowledge models	No	No	Yes	No	Yes

Compared to existing models, our proposed model fully integrates the process of trace engineering. It is platform independent. Its effectiveness lies, first of all, on the fact that it allows both to have a unique model, a trace modeling and an indicator modeling and their process of getting as classes with inheritance relationship, which promotes the reuse of indicators after calculation as full traces. This advantage of our model makes it possible to gain in genericity and reuse of the various elements entering into the calculation of the indicator, thus making the latter easier and faster.

5   Conclusion

In this article, we have presented a theoretical and practical framework for automatically calculating different types of indicators in a Computer Environment for Human Learning (CEHL) context. Our proposed model is based on model- driven engineering (traces of modeled interactions), is independent of the CEHL source of the traces, and favors the reuse of indicators as potential traces for later use or as parameters in the calculation of other indicators. Also, its indicator classification system allows the user to reduce the calculation time of certain indicators by offering the possibility of using the values of certain indicators already calculated on which they depend. This is for example the case of collaborative indicators which most often depend on activity indicators. The remainder of this work will, on the one hand, focus on the best variation of the source platforms of the traces to confirm the independence of the CEHL module and, on the other hand, the integration of forecast indicators will be discussed. For this, it will be discussed to integrate this model a machine module Learning, which offer has the user (teacher) with information to be able to predict certain behavior to come as the likely failure of a learner and to be able to take appropriate action to prevent the situation from arising.

6   References

[1] Settouti, L.S. (2006). Trace-based systems for human learning, 1st Young Researchers' Meeting in EIAH, RJC-EIAH, pp 131, 139.

[2] Gendron E. (2010). Conceptual framework for the elaboration of collaboration indicators from activity traces. PhD thesis report, University Claude Bernard Lyon1.

[3] Pham thi ngoc D. (2008). Re-engineering HIE: automating and re-using usage analysis know-how. [Original title : Réingénierie des EIAH : automatiser et réutiliser le savoir-faire en analyse d'usage]. 2nd meeting of young researchers RJC-EIAH08, Lille, France, p.99-104.

[4] Champin, P.A. (2017). Systems based on modelled traces. [Original title : Systèmes à base de traces modélisées]. Dynamics Knowledge course, Master IA – Université Lyon 1, 2017.

[5] Diagne, F. (2009). Instrumentation of supervision through indicator reuse: Models and Architecture. [Original title : Instrumentation de la supervision par la réutilisation d'indicateurs: Modèles et Architecture]. PhD thesis, University Joseph Fourier Ŕ Grenoble.

[6] Settouti, L.S, Prié, Y., Mille, A., and al. (2006). Trace-based systems for human learning, International Symposium ICTE 2006. "Information and Communication Technologies in Higher Education and Enterprise. INP Toulouse, France.

[7] Tarek Djouad. (2011). Engineering activity indicators from modelled traces for an in-computer human learning environment. PhD thesis, Université Claude Bernard - Lyon I, 2011.

[8] Dimitracopoulou, A., Bruillard, E. (2006). Enriching forum interfaces by visualising automatic analyses of interactions and content. [Original title : Enrichir les interfaces de forums par la visualisation d'analyses automatiques des interactions et du contenu]. Information and Communication Sciences and Technologies for Education and Training, ATIEF, 2006, pp13, 40. https://doi.org/10.3406/stice.2006.942

[9] Choquet, C., Iksal, S. (2007). Modelling and construction of traces of use of a learning activity: a language approach for the reengineering of a HIE. Information and Communication Sciences and Technologies for Education and Training. [Original title: Modélisation et construction de traces d'utilisation d'une activité d'apprentissage : une approche langage pour la réingénierie d'un EIAH. Sciences et Technologies de l'Information et de la Communication pour l'Éducation et la Formation], ATIEF, 2007, pp14, 24. https://doi.org/10.3406/stice.2007.968

[10] Lebis, A. (2019). Capitalizing on the learning outcomes analysis process: ontological modelling & reuse support. [Original title: Capitaliser les processus d'analyse de traces d'apprentissage : modélisation ontologique & assistance à la réutilisation]. Computer Environments for Human Learning. Sorbonne Université, CNRS, Computer Science Laboratory of Paris 6, LIP6, Paris, France, 2019. https://doi.org/10.7561/sacs.2018.2.161

[11] Kris, L., Mille, A. (2009). Traces, interaction traces, learning traces: definitions, computer models, structuring, processing and uses, Trace analysis and personalisation of computer environments for human learning. [original title: Traces, traces d'interactions, traces d'apprentissages : définitions, modèles informatiques, structurations, traitements et usages, Analyse de traces et personnalisation des environnements informatiques pour l'apprentissage humain]. Hermes Sciences Publications, 2009, pp. 21-66. https://doi.org/10.4000/traces.3743

[12] P.-A. Champin, Y. Prié and A. Mille. (2004). MUSETTE: a Framework for Knowledge Capture from Experience. 4th, Knowledge Extraction and Management days

[13] Settouti, L.S, Prié, Y., Mille, A., and al. (2007). Towards model-based systems for HIEs, Oct 2006, LIRIS Lyon, France, 2007.

[14] xAPI Specification. URL : https://github.com/adlnet/xAPI-Spec/blob/master/xAPI-About.md (visité le 12/10/2019).

[15] Yan-Hong, S. (2018). Design of Digital Network Shared Learning Platform based on SCORM Standard. iJET Journal, Vol13, No 07, PP 215-216. https://doi.org/10.3991/ijet.v13i07.8602

[16] Nissrin N., Noura A. (2021). A Proposed Learner's Data Model: Integrating Informal Learning and Enhancing Personalization and Interoperability. iJET Journal, Vol 16, No 08, PP 176-177, https://doi.org/10.3991/ijet.v16i08.19833

[17] H. Slimani, N. El Faddouli, S. Bennani, N. Amrous.(2016). Models of Digital Educational Resources Indexing and Dynamic User Profile Evolution. iJET Journal, https://doi.org/10.3991/ijet.v11i01.4932

7 Authors

Dr Bernabé BATCHAKUI is a senior lecturer at the National Advanced School of Engineering of the University of Yaoundé 1 (UY1), PO Box 8390 Yaoundé-Cameroon. He is interested in artificial intelligence applied to education with a strong focus on knowledge engineering for the resolution of problems in developing countries (especially the problems of the digital divide and access to learning resources in a context of massification). He is a reviewer for high profile conferences and journals: Educon, Cari, CRI, ... His works has resulted in many publications in the e-learning.

Pr. Thomas Djotio is Associate Professor, PhD-Eng at the National Advanced of Engineering of the University of Yaoundé 1 (UY1), PO Box 8390 Yaoundé-Cameroon; and Guest Professor at the University of Applied Sciences Bonn-Rhein-Sieg (H-BRS), Germany. He is currently the principal scientific investigator of NetSAp (Network Services & Applications). He was Alexander von Humboldt Fellow from 2015 to 2017, working on the WiABox project; a multiband wireless access technology to be coupled WiBACK technology. Currently, he is working on the Wireless Community Networking, Applications and Services (WiCoNAS) project, for design and providing content and proximity services on top of the WiABox-WiBACK infrastructure for the local communities. The Network Infrastructure, Applications, Services and Education, are the building stones of his research works.

Dr Ibrahim Moukouop is a Senior Lecturer in Applied Mathematics at the National Advanced School of Engineering of the University of Yaounde 1, PO Box 8390 Yaoundé- Cameroon. He is also a computer science engineer with 20 years of experience, in the field of software engineering and software testing. He published many articles and communications in the fields of e-learning, software engineering, software security and applied mathematics.

M. Alex Ndouna is PhD student in Computer Engineering at the National Advanced School of Engineering of the University of Yaounde 1, PO Box 8390 Yaoundé- Cameroon. He is also a computer science engineer with 5 years of experience.