An MDA-based Approach for the Design and Automatic Computation of Collaboration Indicators in E-Learning Systems

Sara Ghaoui
ICOSI Laboratory, Abbes Laghrour University, Khenchela, Algeria
sara.ghaoui@univ-khenchela.dz (corresponding author)

Sofiane Mounine Hemam
Abbes Laghrour University, Khenchela, Algeria | National High School of Cyber-Security, Algiers, Algeria
hemam.sofiane@univ-khenchela.dz

Tarek Djouad
Abbes Laghrour University, Khenchela, Algeria
tarek.djouad@univ-khenchela.dz

Received: 18 February 2025 | Revised: 3 April 2025 | Accepted: 9 April 2025

 DOI: https://doi.org/10.48084/etasr.10607

ABSTRACT

Effectively assessing collaboration in e-learning environments requires robust indicators that capture learner interactions. However, most existing approaches are platform-dependent, rely on predefined indicators, and offer little flexibility for tutors and teachers. This limits their ability to tailor assessments to specific learning contexts. To address these challenges, this study proposes a Model-Driven Architecture (MDA)-based approach that allows tutors and teachers to design and compute collaboration indicators independently of any specific platform, without requiring advanced technical expertise. This approach introduces a flexible and scalable framework for the design and computation of collaboration indicators in e-learning analytics. First, it ensures platform-independent computation by leveraging a model transformation approach within an MDA-based process. Second, it provides a formal model that enables teachers and tutors to define meaningful and adaptable indicators that align with their specific observation needs. Finally, the proposed DECIN-AGST system automates the generation of transformation sequences. The accuracy and reliability of the proposed system were validated through formal verification using the UPPAAL model checker.

Keywords-collaboration indicator; e-learning system; sequence of transformations; MDA; learning assessment; e-learning analytics

I. INTRODUCTION

Collaborative e-learning is a distance learning process that uses e-learning systems [1]. In this type of learning, each learner actively works to build his knowledge through exchange and collaboration with group members [2]. Recent studies have emphasized the importance of multimodal learning analytics in predicting student collaboration satisfaction, offering insights into engagement and participation levels [3]. Communication barriers and coordination difficulties have been identified as key obstacles to online collaboration [4]. Despite these challenges, experts emphasize the benefits of integrating learning analytics with collaborative learning, highlighting the need for more flexible and scalable solutions [5]. In this context, the evaluation of online collaborative work remains one of the current limitations of collaborative distance learning [6], as it involves not only evaluating the acquisition of knowledge on an individual level but also measuring the participation of each learner or group of learners in collaborative work [7]. The construction of the evaluation grid in an e-learning system is based on the verification of knowledge acquisition and the observation of the behavior and quality of the work method [8]. These observations are used to define an indicator [9], which is a variable in the mathematical sense that provides information based on observation. It is a means of evaluation that must serve as an aid in decision-making. The collaboration indicator in an e-learning system provides information about the level of participation and collaboration of learners in the collaborative learning process [10].

During the use of an interactive e-learning system, many actions and events are generated (connection, disconnection, chat, reading, etc.). These can serve as valuable sources of information, called traces. Computing indicators from traces to support collaboration and collaborative learning is a very active field of research [11]. Previously, the description of the elements stored during an e-learning session was limited to textual documentation, which made human exploitation very difficult. Additionally, designing collaboration indicators that can be automatically calculated is a challenge that requires expertise in computer science. Therefore, a teacher without a computer science background cannot perform this task. Furthermore, most current systems for calculating collaboration indicators are closed, which means that these indicators must be designed and defined before the development of the system. When a new collaboration indicator needs to be added, the system must be updated to include it. Thus, existing systems do not take into account the observational needs expressed by the teacher or tutor, limiting them to predefined collaboration indicators. To address these challenges, this paper proposes an approach based on Model-Driven Architecture (MDA) to design and compute collaboration indicators.

In recent years, an important part of the research on e-learning environments has been oriented toward evaluating learners' activities and calculating indicators. In [12], a new computer object, named m-trace (modeled trace), was proposed, which associates each collection of observed elements with a model to formally describe the structure and content of the trace. Thus, the m-trace is a trace associated with its model, and a trace model is a formal description of the structure and content of the trace. It provides information about the properties, the elements it contains, and the relationships between the elements. However, a trace meta-model defines the properties that all traces must satisfy to ensure interoperability between traces. A system based on modeled traces is any computer system that enables the manipulation of m-traces by supporting the management, transformation, and visualization of modeled traces.

More recently, multiple studies have focused on assessing the effectiveness of monitoring indicators in collaborative systems, using metrics such as recall, precision, and f-score to evaluate their performance [13]. Integrating these evaluation metrics has proven valuable in tracking and enhancing learner engagement in online environments, making them essential for adaptive and personalized learning experiences [14]. In [15], it was examined how automated analytics can assess collaboration skills by analyzing group speech data. Natural Language Processing (NLP) techniques and machine learning models were used to analyze communication patterns, detect engagement levels, and identify collaboration dynamics in real time. In the ICALTS project [16], indicators were identified through the analysis of student interactions at the metacognitive level that could allow them to self-regulate or evaluate their activity. In [17], indicators were proposed to evaluate learning activity using discussion forum posts. These indicators were also used to validate the quality of asynchronous discussion without passing the content analysis stage. In [18], a mechanism was developed that included activities to allow students to examine a collaborative task from different angles. A set of metrics-based indicators was proposed to evaluate the group's results as the final product of the collaborative work. In [19], the focus was on calculating collaboration indicators in Moodle, using learning analytics and data mining techniques to define specific collaboration indicators, such as participation rate, interaction frequency, response time, message length, and role distribution in group discussions.

In [20], specific algorithms were proposed to extract and compute collaboration indicators, following a predefined framework structured around multiple dimensions. First, the collaboration indicator "individual participation" is assessed through metrics such as the number of contributions or the length of texts added or modified by each learner. Next, the collaboration indicator "equity of contribution" is measured based on the distribution of actions among group members, using indicators such as entropy or the Gini coefficient to quantify disparities. Furthermore, the collaboration indicator "temporality and collaborative rhythm" is analyzed by examining the intervals between contributions to detect signs of consistent engagement or, conversely, periods of inactivity. Finally, the collaboration indicator "interaction and feedback" is calculated by evaluating the frequency of peer comments or responses to suggestions. These algorithms allow a more detailed and refined analysis of collaborative dynamics, offering a rich perspective on engagement and interactions within learning groups. However, in these works, indicators were often made in an ad hoc manner and were usually designed specifically for a platform with no explicit desire to reuse them.

The second category of existing work aims to provide methods/tools to facilitate the design and calculation of collaboration indicators. The REDiM project [21] focuses more specifically on analyzing the traces collected by a Computer Environment for Human Learning (CEHL) to provide the teacher with calculated indicators. With the same objective and to efficiently design and compute indicators, a method was proposed in [11] to facilitate trace analysis in learning situations based on model-driven engineering. Saving the transformations applied to the traces was proposed to facilitate their reuse. These sequences of transformations made it possible to pass from the primary trace model to the indicator model. In [10], a computer tool called Genidic was developed to assist users in indicator development, management, and computation. It used a rule-based system, where the traces were facts and the processes for calculating indicators were rules. In [22], UTL (Usage Tracking Language) was proposed to allow the definition of indicators in a form similar to design patterns, designed to address capitalization and reuse issues. However, UTL did not initially have the tools to formally specify how to calculate an indicator from the collected traces. The textual descriptions in UTL to produce an indicator from the traces do not allow for the automatic generation of indicator values. In [23], a new version of UTL called DCL4UTL was proposed, which allows indicators to be modeled in a form that can be capitalized, automated, and reused to provide significant indicators to the teacher or tutor. In [24], a multi-agent system based on fuzzy logic was proposed to allow evaluation of the level of learning collaboration. The inputs of the fuzzy system are the calculated indicators, which are based on the analysis of learner traces. In [25], a Learning Management System (LMS) was proposed based on cloud architecture by integrating a multi-agent system to collect, analyze, and filter traces to calculate interaction indicators that favor collaboration. Table I summarizes and compares these works with the proposed approach.

TABLE I. COMPARISON BETWEEN RELATED WORKS AND THIS PROPOSAL

	Supports the design or calculation of indicators	**Indicator type**	**Used approach**
[22]	Design	Indicators in CEHL	Designing of indicator in a form similar to a Design pattern.
[11]	Calculation	Indicators in Moodle	- Based on MDE. - Platform-oriented model transformation. - Using an m-trace-based system (self-developed).
[10]	Design and calculation	Collaboration indicators	- Using a collaboration indicator pattern for the design. - Using an Artificial Intelligence-oriented computation method, based on rule logic. - Using a rule-based System (self-developed).
[23]	Design and calculation	Indicators in CEHL	- Enrich UTL with formal language to formally describe the calculation method. - Indicator computation using a DCL4UTL interpreter (self-developed) - Integrated into a trace analysis tool.
[24]	Calculation	Collaboration indicators in collaborative e-learning systems	Ad-hoc manner
[25]	Calculation	Collaboration indicators in e-learning systems	- Model transformation oriented. - Based on MDE. - Using a multi-agent system and an m-trace-based system (KTBS).
This work	Design and calculation	Collaboration indicators in e-learning systems	- Design based on the DCIN model. - Calculation oriented at model transformation based on MDA. - Transformation sequences are automatically generated using DCIN-AGSET (self-developed).

These works offer a variety of contributions. In [22], the design of indicators in CEHL was proposed in a form similar to a design pattern. However, this method initially lacked the means to formally specify how to compute the indicator from the collected traces. In [11], an MDE-based method was proposed to compute indicators on a specific platform. In [25], the same method was used to create a multi-agent system to compute collaboration indicators in an e-learning system. These last two works focus on the computation method rather than indicator design, and the proposed systems are closed, i.e., the teacher must use predefined indicators to assess the behavior of their learners and cannot propose their own indicators. In [10], a method for the design and computation of collaboration indicators was proposed using a collaboration indicator pattern for the design and a computation method oriented at artificial intelligence based on rule logic. In [23], a method for enriching UTL was proposed, using formal language to describe the calculation method. In all previous works, methods for designing and calculating indicators were proposed, where the indicator designer must necessarily be a computer scientist, whereas the non-computer scientist teacher who wants to evaluate the behavior of their learners in a learning session cannot propose their own indicators. Thus, this study aimed to develop a platform-independent method for designing and calculating collaboration indicators, where a non-computer scientist teacher can design and calculate his own collaboration indicators with minimal intervention from a computer scientist. Compared to the above works, the proposed one considers the design and computation of collaborative indicators in e-learning systems. Therefore, its contributions can be summarized as follows:

- Focuses on the calculation of collaboration indicators in e-learning systems independently of any platform used. For this purpose, a model transformation approach and an MDA-based process are proposed to obtain collaboration indicator models.
- Focuses on the design of collaboration indicators in e-learning systems. For this purpose, a formal model is proposed to facilitate the design of valid and meaningful collaboration indicators according to the teacher's observation needs.
- Based on the application development process supported by MDA, the calculation of the collaboration indicator can be seen as a model transformation process, where the trace model is passed through a sequence of transformations to arrive at the collaboration indicator model. For the same previous need for automatic calculation of collaboration indicators and the acquisition of specific computer skills that cannot be achieved by a non-computer scientist teacher, an automated sequence of transformations is proposed. This focuses on how to obtain sequences of transformations by proposing a system that ensures their automatic generation is applied in the m-trace base system to arrive at the indicator model.

II. PROPOSED APPROACH

The following three objectives should be achieved. The first is to propose a method that allows us to design and compute collaboration indicators independently of any platform. For this purpose, MDA is considered a theoretical and technological framework to base the proposal, which includes two types of models: the initial models created by the designers and the derived models obtained automatically. Concerning the initial models to create, two Platform-Independent Models (PIM) are proposed. The first is a meta-model of trace in e-learning systems (T-PIM), where any trace model specific to an implementation platform (T-PSM) must conform to it. The second is a generic meta-model of collaboration indicators in e-learning systems (CI-PIM). Thus, any collaboration indicator model must comply with this CI-PIM. However, the Trace Platform-Specific Models (T-PSM) are initially created by the designer, and the Collaboration Indicator models (CI-PSM) are derived, obtained automatically by applying a sequence of transformations to the Trace Model (T-PSM). OCL was used to specify the transformation contracts and to ensure that the obtained CI-PSM is the valid result of the transformation.

The second objective is to propose the DCIN model (Design Collaboration Indicators Model) for the design of collaboration indicators in an e-learning system. It is especially intended for non-computer specialist teachers who want to evaluate learners' collaboration following collaborative distance learning. The model allows for the proposal of valid indicators, either by respecting the observation needs of the teacher or by following the predefined paths associated with the model to obtain predefined collaboration indicators. The DCIN model is in the form of a timed automaton [26], which is a finite-state automaton with a finite set of real-time clocks. This model will be integrated into a real-time collaboration indicator design system.

The third objective is to automate the transformation process for obtaining CI-PSMs. DCIN-AGST is proposed for this purpose, which is a formal system for the Design of Collaboration Indicators and Automatic Generation of Sequences of Transformation. DCIN-AGST is a real-time system modeled as a network of timed automatons. The DCIN model will be integrated into the DCIN-AGST formal system.

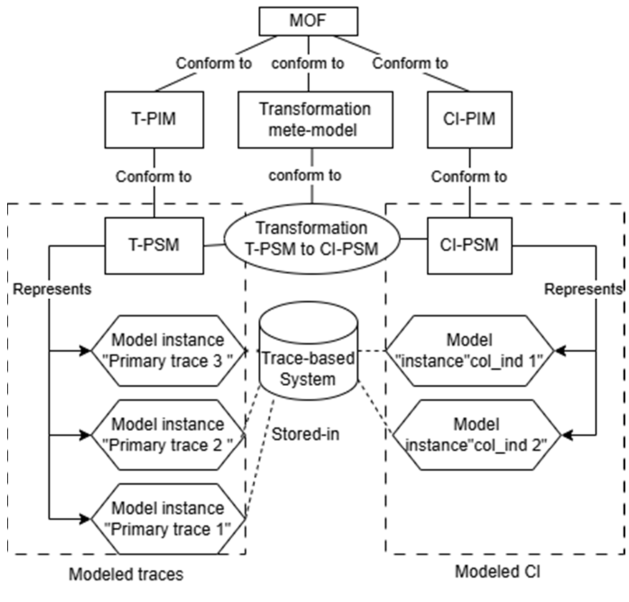

Fig. 1. MDA-based approach to calculate collaboration indicators.

A. Proposed Approach Based on MDA

MDA is one of the most famous implementations of model-driven engineering. It focuses on the transformations made to the models, and not on the system's source code, which greatly reduces the effort of designers, teachers, and researchers. All the work is based on the transformations of the models that prescribe the systems and not the systems themselves [27]. The proposed approach is based on MDA, where the calculation of collaboration indicators can be seen as a set of model transformations, starting with a trace model and arriving at the collaboration indicator model. From the latter, a simple function call is enough to obtain the final value of the collaboration indicator. According to the MDA principles, each PSM model must conform to a PIM model, which describes the system independently of any technical platform and any technology used to deploy the application. In this approach, as presented in Figure 1, each trace model specified for an e-learning platform (T-PSM) must conform to a trace meta-model (T-PIM) that describes the trace structure independently of any e-learning platform to be used.

In the same sense, each CI-PSM must conform to a CI-PIM. MDA supports all development steps and standardizes the transitions between them. The first step is to create PIMs and enrich them successfully. The second step is to choose the implementation platform and generate the PSMs. This MDA-based approach proposes a T-PIM that serves as a meta-model for collaborative and non-collaborative traces. In addition, a collaborative indicator platform-independent model is proposed to represent a meta-model (CI-PIM) in e-learning systems. As shown in Figure 1, both models conform to the OMG Meta Object Facility (MOF) standard [19]. A sequence of transformations is a series of operations that takes a T-PSM input and provides a CI-PSM output. Obtaining the CI-PSM is a process based on a sequence of model transformations. These transformations must be written in a transformation language (transformation meta-model). This study uses a simplified version to be translated according to the transformation language implemented in the trace-based system to be used. CI-PIM is associated with OCL constraints to express requirements that cannot be expressed with UML. These constraints are used to confirm the transformation process.

1) Trace Platform Independent Model (T-PIM)

T-PIM serves to model the collaborative/non-collaborative trace independently of any platform used. Any T-PSM must conform to this meta-model. As shown in Figure 2, a T-PSM represents a primary trace, which is a projection of a specific e-learning system and conforms to the T-PIM. Figure 3 shows T-PIM as a UML class diagram. T-PIM distinguishes 13 classes, which are defined in Table II.

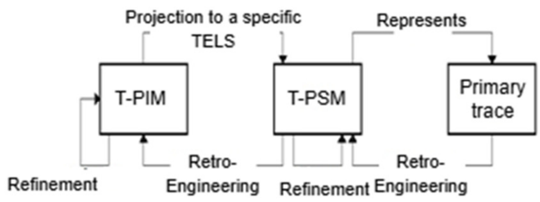

Fig. 2. The relationship between trace levels.

The following rules are proposed in the meta-model:

- Each trace has a starting date and an ending date and is made up of a series of observed elements (obsels).
- An obsel (action) can be collaborative, for example, in the case of sending a message, or non-collaborative, such as a disconnection action.
- An obsel can be associated with a tool.

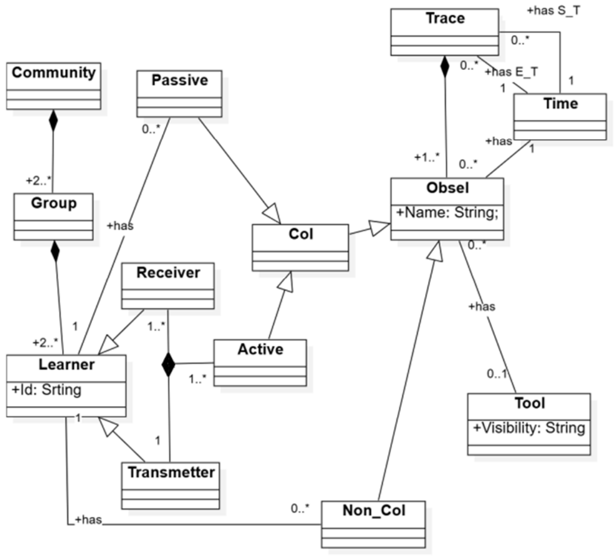

Fig. 3. Trace platform independent model (T-PIM).

TABLE II. T-PIM CLASSES

Class	Description
Obsel	The observed element (obsel) class represents the action to observe, e.g., sending a message, consulting a wiki, logging in, etc.
Col	Represents collaborative action, for example, participating in a survey.
Non-collaboratif	Represents a non-collaborative action, such as consulting a course.
Active	Represents a send action or active participation.
Passive	Represents a consultation action.
Learner	Associated with the obsel: it is the user who performs the action.
Transmitter	Represents the actor who does the action in an active collaborative action.
Receiver	Represents a non-empty set of learners concerned by the act of transmitting.
Time	Associated with the obsel - This is the action time.
Tool	Associated with the obsel. Represents the tool that supports the interaction. It can be public or private (private zone tool for each group).
Group	Represents a set of learners.
Community	Represents a set of groups.
Trace	Represents a trace in an e-learning system.

- Each obsel has a start and end date.
- A collaborative obsel can be passive if it is a consultation action, e.g., consulting a forum, or active if it is an action of active participation in the learning process, e.g., editing a wiki or sending a message.
- A passive collaborative obsel is associated with a single learner.
- An active collaborative obsel is associated with a transmitter and a set of receivers.
- A learner group is composed of two or more learners.
- A community can be made up of one or more groups.

2) Collaboration Indicator - Platform Independent Model (CI-PIM)

CI-PIM serves to model the collaboration indicators independently of any platform used. Any CI-PSM to be obtained must conform to this meta-model. As shown in Figure 4, a CI-PSM represents collaboration indicators in an e-learning system, which is a projection of a specific e-learning system and conforms to the CI-PIM. Figure 5 shows the CI-PIM as a UML class diagram.

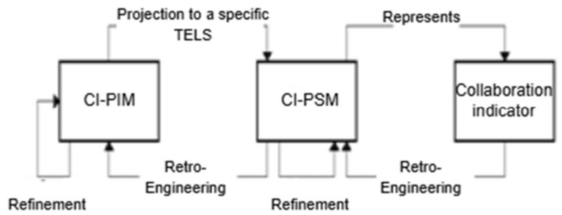

Fig. 4. The relationship between collaboration indicator levels.

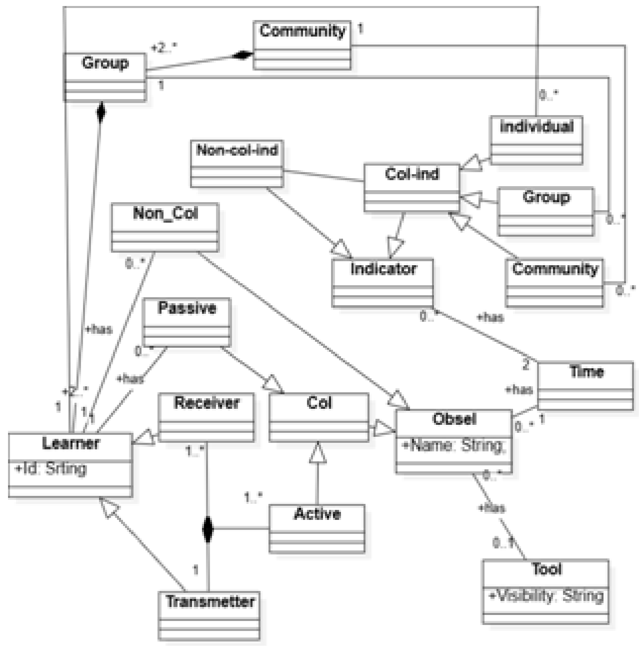

Fig. 5. Collaboration Indicators Platform Independent Model (CI-PIM).

To avoid repetition, only classes not mentioned in T-PIM will be defined, as shown in Table III.

TABLE III. THE CI-PIM CLASSES DESCRIPTION

Class	Description
Indicator	Represents the indicator in e-learning systems.
Col-Ind	Represents the collaboration indicator in an e-learning system.
Non-Col-Ind	Represents other types of indicators in an e-learning system.
Individual	Represents a collaboration indicator of a learner.
Group	Represents a collaboration indicator of a Group.
Community	Represents a collaboration indicator of a community.

The following rules are proposed in this meta-model:

- An indicator in an e-learning platform can be elementary or composed of indicators.
- An indicator is calculated within a well-defined time interval. An indicator can be:
  - Non-collaborative, which is associated with non-collaborative obsels. For example, the R-time indicator equals the average time taken to complete a course.
  - Collaborative, which is associated with collaborative obsels. For example, a learner's rate of participation in collaborative work within his or her group.

A collaboration indicator can be calculated on an individual, group, or community level, e.g., the communication rate indicator for a learner, a group of learners, or a community.

3) Transformation Mode

The model transformation process is a unidirectional, exogenous, horizontal, M2M transformation where:

- It is realized in one direction, from T-PSM to CI-PSM.
- T-PSM and CI-PSM are conforming to different meta-models, which are T-PIM and CI-PIM.
- T-PSM and CI-PSM are at the same abstraction level.
- It is a model-to-model transformation.

The model transformation process is based on an imperative approach where the source model (T-PSM) is browsed in a certain order to generate the target model (CI-PSM). The transformation operators in a trace-based system are divided into two classes. The first class concerns operators that do not modify the source model (TPSM), such as selection and fusion. The second class concerns operators that modify the source model, such as matching, rewriting, and pruning models. The DCIN-AGST formal system uses a simplified version of the generated transformations to be translated according to the transformation language implemented in the trace-based system to be used.

a) Transformation Contracts

To express requirements that cannot be articulated using UML and to confirm the transformation process, the following constraints are proposed in OCL, as shown in Table IV.

B. The DCIN Model For The Design of Collaboration Indicators

The proposed DCIN model is based on works that deal with collaboration indicators. Each collaboration indicator has a name and an informal description [10], and according to [28], it can be calculated at the individual, group, or community level, and can be elementary or composite [10]. An elementary indicator is constructed from traces on which a set of transformations and computations are applied to arrive at the indicator value, whereas a composite indicator is obtained from other elementary indicators. Thus, from the above definitions of a collaboration indicator, and as shown in Figure 6, the DCIN model facilitates the design of collaboration indicators at the individual, group, and community levels. The collaboration indicators proposed in this model can be elementary or composite. Timed automata for the collaboration indicator design facilitate this process by requiring the designer to identify the indicator's name, its informal description, and its time interval. The designer must select the appropriate type of indicator, either elementary or composite. Then, regardless of his or her previous choice, he or she must define the set of obsels involved. After that, he or she will decide whether the indicator is personal, group, or community-based. The model contains predefined paths that can be seen as ready-to-use patterns for designing collaboration indicators in an e-learning system. These paths allow for the design of collaboration indicators of the types "number" and "rate". A path represents an automaton's execution [26]. The paths in this model are valid collaboration indicators.

TABLE IV. OCL CONSTRAINTS

Informal constraint	OCL constraint
In an active collaboration situation, the set of receivers must be non-empty and different from the emitter.	Context Active Inv: (This.Receiver .allInstance()) →not-empty and forAll (p|p <> this.Transmitter)
A collaboration indicator should be calculated in relation to at least one collaborative obsel.	Context Col-ind Inv: Col-ind.allInstance() forAll(p1| p1.obsel.Col→not-empty)
The starting date of obsels related to the indicator must be greater than or equal to the indicator's starting date.	ContextIndicator Inv: Indicator.Starting.time<= Indicator. obsel.Starting time
The end date of obsels related to the indicator must be less than or equal to the indicator's ending date.	Context Indicator Inv: Indicator.Endtime>=Indicator.obsel.End time
The indicator start date must be less than or equal to the indicator end date.	Context Indicator Inv: Self.End time>= Self.obsel.Starting-time
The obsel start date must be less than or equal to the obsel end date.	Context Obsel Inv: Self.End time>= Self.obsel.Starting time
Each learner (obsel actor) must have a distinct identifier.	Context Learner Inv: Learner.allInstances()→forAll (p1,p2|p1<>p2 implies p1.id<> p2.id)
A learner must be a member of only one group	Context Learner Inv: Learner.Group→size ()=1
A learner must be a member of only one community.	Context Learner Inv: Learner.Group.comumunity→size ()=1
Each group must have a unique identifier	Context Group Inv: Group.allInstances()→forAll (p1 ,p2 |p1<>p2 implies p1.id <>p2.id)
Each group should contain a set of unique learners	Context Group Inv: Group.Learner.allInstances()→ forAll(p1,p2|p1<>p2)
Each community should contain a set of unique groups.	Context Community Inv :Community.Group.allInstances()→forAl l(p1,p2 | p1<>p2)

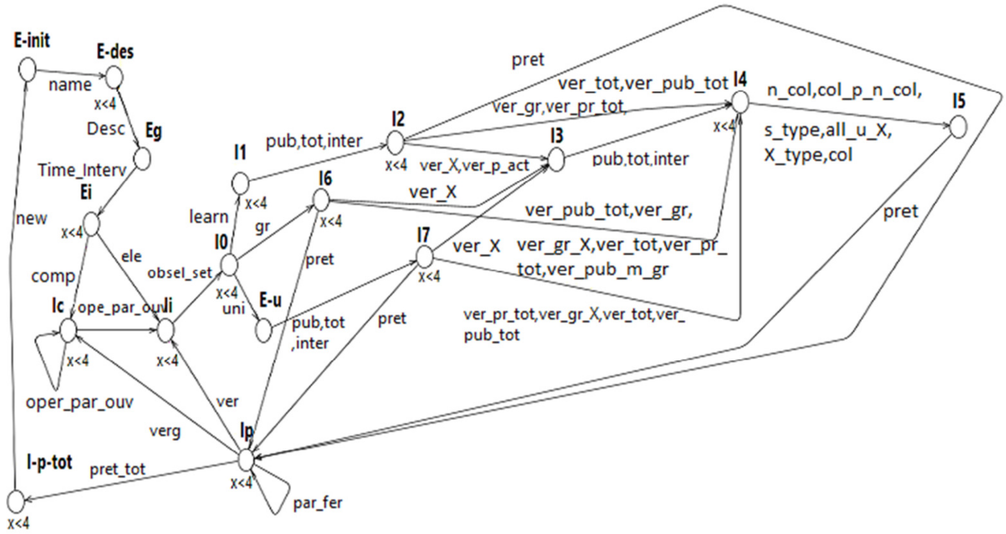

Fig. 6. DCIN model.

1) Formal Definition of the DCIN Model

The model is in the form of a timed automaton, such that E represents a finite set of transition labels, S represents a finite set of places, X = (x) is a finite set of variables with a value in R^+ (clocks), and I: S → C(X) is a function that maps to a state s the set of conditions on the square called invariant. T⊂ S×E×C(X) × 2^x × S is the set of transitions. A transition e ≤ s, a, k, m, s' ∈ T represents the transition from place s to place s', guarded by constraint k, labeled by a, and resets the variables m ∈ X.

E–init belongs to S and is the initial place. The elements of the sets E and S are:

S = (E–init, E–des, Ei, Eg, Ii, I0, I1, I2, I3, I4, I5, I6, I7,Ec, Ip, I–p–tot).

E = (ver_gr, ver_pub_m_gr, col_p_n_col, Desc, Time_Interv, ver_pub_tot, obsel_set, ver_gr_X, ver_X, ver_pr_tot, ver_p_act, all_u_X, ver_tot, n_col, Name, Uni, Inter, X_type ,Col, s_type, oper_par_ouv, ope_par_ouv, Ele, pret_tot, pret, Learn, ver, verg, gr, pub, par_fer, tot).

The elements of set E are defined in Table V. The following examples illustrate three types of collaboration indicators that can be designed using this model.

The first example concerns the elementary indicator "Nbr chat-gr," which is equal to the number of chat messages from a learner to the members of his group. To design this indicator, the designer will follow the following path:

$$(E_{init},0) \rightarrow_{name} (E_{des},0) \rightarrow_{Des} \left(E_g,0\right) \rightarrow_{Time-interv}$$
$$(E_i,0) \rightarrow_{ele} (I_i,0) \rightarrow_{obsel-set} (I_0,0) \rightarrow_{learn} (I_1,0)$$
$$\rightarrow_{inter} (I_2,0) \rightarrow_{pret} (I_p,0) \rightarrow_{pret-tot} (I_{p-tot},0).$$

The second example concerns the composite indicator "group participation rate," which equals the number of collaborative interactions within the group, divided by the total number of collaborative interactions in the e-learning system. To design this indicator, the designer will follow the following execution:

$$(E_{init},0) \rightarrow_{name} (E_{des},0) \rightarrow_{Des} \left(E_g,0\right) \rightarrow_{Time-interv}$$
$$(E_i,0) \rightarrow_{compo} (I_c,0) \rightarrow_{ope-par-ouv} (I_i,0) \rightarrow_{obsel-set}$$
$$(I_0,0) \rightarrow_{uni} (E_c,0) \rightarrow_{tot} (I_7,0) \rightarrow_{pret} (I_p,0) \rightarrow_{par-fer}$$
$$(I_p,0) \rightarrow_{pret-tot} (I_{p-tot},0).$$

TABLE V. ELEMENTS OF THE SET E

Transition label	Description
Ver_gr	Relative to the private activities of the members of his group.
Ver_pub_m_gr	Relative to the public activities of group members
Col_p_n_col	Collaborative + non-collaborative activities
Desc	Textual description of the indicator to be designed.
Time_Unterv	Time interval of the indicator to be designed.
Ver_pub_tot	Compared to the total public activities.
Obsel_set	The set of observed elements concerned.
Ver_gr_X	Relative to the activities of a group X.
Ver_X	Relative to the activities of a learner
Ver_pr_tot	Relative to the total private activities.
Ver_p_act	Relative to the learner's activities.
All_u_X	All obsel types except initial types.
Ver_tot	Relative to the total activities.
n_col	Non-collaborative activities
Inter	Orivate within the group
X_type	All obsels of type X.
comp	Compound indicator
Col	Collaborative activities.
s_type	Same initial type.
oper_par_ouv	operator (.
ele	Elementary indicator.
pret_tot	Ready totally
pret	ready partially
learn	of a learner
ver	comma",".
gr	Of a group.
pub	Public.
par-fer	) .
tot	Total.

The third example concerns the usage of a predefined path to design the indicator "rate of expression of ideas in an e-learning system," which is equal to the number of total forum publications divided by the number of total collaborative interactions. To design this indicator, the designer can follow the below path:

$$(E_{init}, 0) \rightarrow_{name} (E_{des}, 0) \rightarrow_{Des} (E_g, 0) \rightarrow_{Time-interv} (E_i, 0) \rightarrow_{ele} (I_i, 0) \rightarrow_{obsel-set} (I_0, 0) \rightarrow_{uni} (E_c, 0) \rightarrow_{tot} (I_7, 0) \rightarrow_{ver-tot} (I_4, 0) \rightarrow_{col} (I_5, 0) \rightarrow_{pret} (I_p, 0) \rightarrow_{pret-tot} (I_{p-tot}, 0).$$

Regarding the design of these collaboration indicators, the following two points must be pinpointed:

- Temporal transitions were ignored for complexity reasons, assuming that the designer makes instantaneous choices.
- The operators in the "composite" type collaboration indicators are written in prefixed notation.

2) Implementation and Functioning of the Proposed Approach

a) DCIN-AGST Formal System

The DCIN model was implemented in a real-time system modeled as a network of timed automatons in a formal system (DCIN-AGST system) under UPPAAL [29]. This DCIN-AGST formal system facilitates the design of collaboration indicators and allows the automatic generation of sequences of transformations. The UPPAAL simulator is used to design the collaboration indicators. The simulation control panel in Figure 7 proposes the possible transitions to be refreshed at each step. The designer must choose one of the transitions. By refreshing the pret-tot transition, the designer finishes the design of his/her collaboration indicator. Then, he/she can retrieve the sequence of transformations and the details of his or her collaboration indicator via the message sequence chart panel in Figure 8.

a) Use Case of the DCIN-AGST Formal System

Let us take the indicator Ic as the number of chat messages sent by a learner to the members of their group. As shown in Figure 9, to design this collaboration indicator, the designer has to determine the name of the collaboration indicator (Ic1), its description, and its time interval. Since Ic1 is elementary, the designer has to refresh the transition "ele." Then, he must select the set of observed elements affected by this indicator (send a message). Next, they refresh the transition "learn" to identify the learner sender. Finally, they refresh the transition "inter" to specify that Ic1 is a collaboration indicator calculated at the group level. By refreshing the transitions "pret" and "pret-tot," the designer completes the design of the collaboration indicator Ic1. The designer can retrieve the automatically generated sequence of transformations, as detailed in the previous section, through the UPPAAL simulator's message sequence chart panel. The left-hand segment shows the design details of the Ic1 collaboration indicator (Figure 9), and the right-hand segment displays the sequence of transformations to be executed (Figure 10).

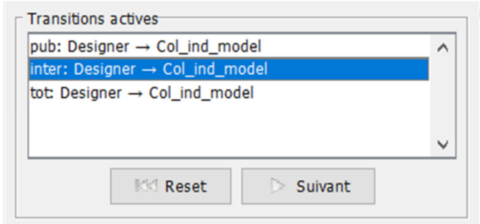

Fig. 7. The simulation control panel of the UPPAAL simulator.

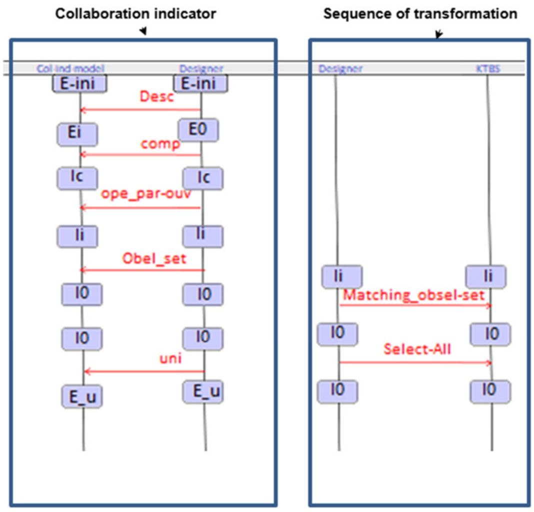

Fig. 8. Message sequence chart panel of the UPPAAL simulator.

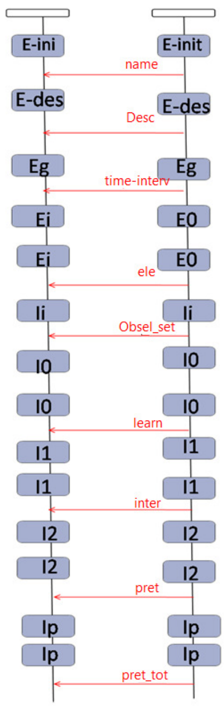

Fig. 9. Execution of the design of the Ic1 indicator.

The DCIN-AGST formal system's automatically generated sequence of transformations allows passage from the trace model to the collaboration indicator model Ic1. As shown in

Figure 10, first, a time interval based on the designer's preference is selected. Then, it is pruned to retain only the relevant obsel sets (send a message). Following this, learner-actor actions are selected. Finally, only the send message actions within the learner's group are kept.

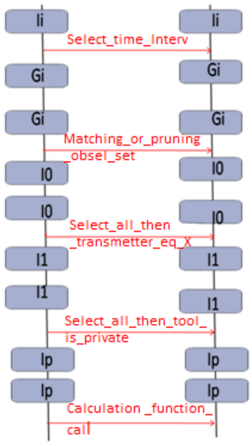

Fig. 10. The automatically generated sequence of transformations for indicator Ic1.

3) Calculation of the Collaboration Indicator

As shown in the sequence diagram (Figure 11) and the flowchart (Figure 12), to calculate a collaboration indicator, the designer must first propose a model of collaborative and non-collaborative traces (T-PSM) in the chosen e-learning system that conforms to T-PIM. Then, the necessary data for constructing the trace base are collected in the m-trace base system. This system is based on the proposed model from the raw traces obtained during the collaborative learning sessions using an e-learning system. Once the trace base is constructed, the designer uses the proposed DCIN-AGST formal system to propose and design a valid collaboration indicator according to the choice of the designer (elementary or compound collaboration indicator; individual, group, or community) by respecting the observation needs of the teacher designer or following one of the predefined paths in the model. Then, the sequence of transformations is retrieved, which are automatically generated by the system. By applying the sequence of transformations obtained in the m-trace base system, the learner can switch from the trace model to the collaboration indicator model, while collecting in parallel the observed elements of the primary trace according to the collaboration indicator model obtained. Finally, they call the computation functions to obtain the final value of the indicator.

III. FORMAL CHECKING FOR THE SYSTEM

The DCIN-AGST formal system is formally verified using the formal verification method "model checking" via the UPPAAL model checker [29]. This allows us to verify the model's safety, reachability, non-blocking/interlocking, and liveliness properties. The verification properties of the system are written in a simplified version of untimed branching time logic (CTL) [30] and Timed Computation Tree Logic (TCTL) [31]. Figure 13 shows the UPPAAL verifier interface, where satisfied properties are marked in green and violated properties in red.

- Non-blocking/interlocking property: A state is a deadlock if there are no outgoing action transitions or delay successors. This property can be expressed as *A[] not deadlock*. The property is satisfied in this system.

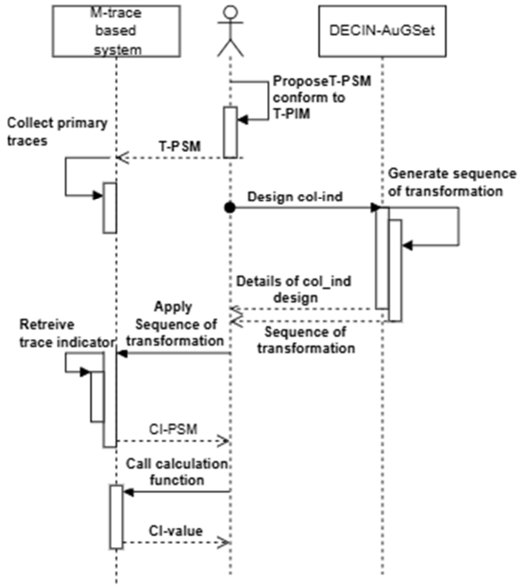

Fig. 11. Sequence diagram showing the steps required to calculate a collaboration indicator.

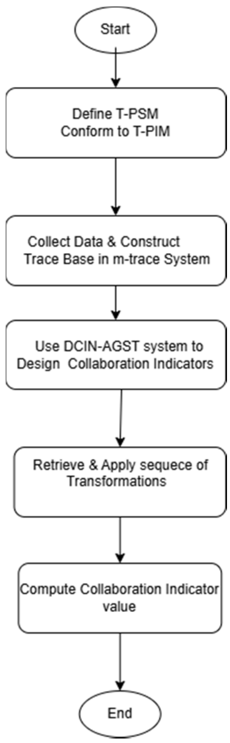

Fig. 12. Flowchart of the collaboration indicator computation process.

- Liveliness property: A desirable event will necessarily happen. For all model paths, it is finally managed to design a valid indicator from the initial state to the fully ready state: *A<>Col-ind-model.Eini imply Col-ind-model.I-p-tot.* The property is satisfied in this system.

- Safety property: An unwanted event has never occurred during the design process. At some point, the designer manages to finish the indicator design, and the model arrives at another state different from I-p-tot (lack of synchronization): *E[] Designer.I-p-tot and not Col-ind-model.I-p-tot.* This property is not satisfied in the system.

- Reachability property: A certain situation should be reached if the designer has chosen to design an indicator of a learner, which must reach one of the final states I5 or I2: *A<>Designer.I1 imply (Col-ind-model.I5 or Col-ind-model.I2).* This property is satisfied in the system.

- Safety property: The model allows us to design an invalid indicator with several opening parentheses greater than the number of closing parentheses: *A<>Col-ind-model.I-p-tot and Col-ind-model.z>0.* This property is not satisfied in the system.

- Safety property: For complexity reasons, the number of compositions must be limited in a composite indicator to avoid entering an infinite loop (the model allows us to do this). The system allows us to propose a finite number of compositions: *A[] Col-ind-model.a<12*. The property is satisfied in the system.

- Safety property: There is at least one path to design a valid indicator without passing through one of the final model states. For example, this property ensures that the indicator number of chat messages or the indicator number of wiki edits cannot be proposed because it is an invalid indicator: *E[] (Col-ind-model.Ip and not Col-ind-model.I2 and not Col-ind-model.I5 and not Col-ind-model.I7 and not Col-ind-model.I6).* The property is not satisfied in the system.

- Safety property: The indicator design must be within a limited time (critical resource problem considering a single model instance): *A[] (Col-ind-model.x1<80 and Designer.x1<80).* This property is satisfied in the system.

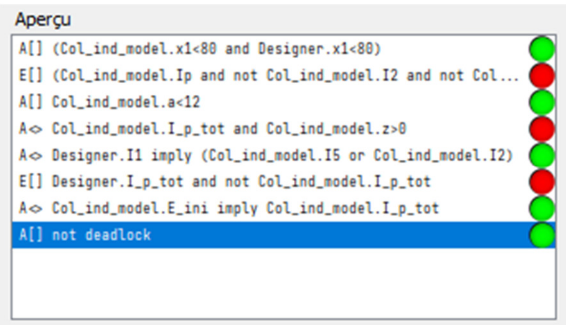

Fig. 13. UPPAAL verifier.

IV. CONCLUSION

This study presented an MDA-based approach to design and compute collaboration indicators in an e-learning system. To achieve this, the DCIN-AGST formal system was proposed, which guides teachers and tutors in designing collaboration indicators. The system allows them to make optimal decisions, thereby enhancing and stimulating collaborative learning.

The Trace Platform Independent Model (T-PIM) and the Collaboration Indicator Platform Independent Model (CI-PIM) were introduced to design and compute collaboration indicators. These meta-models are platform-independent, ensuring flexibility across different e-learning systems. Any trace model specific to an e-learning system (T-PSM) must conform to T-PIM. To derive a collaboration indicator model (CI-PSM), a series of model transformations must be applied, converting the T-PSM into the CI-PSM, which must conform to the proposed CI-PIM. The sequence of transformations is automatically generated through the DCIN-AGST formal system, which has been formally verified using model checking with the UPPAAL model checker.

Compared to existing works, this approach presents several key advancements:

- Custom Collaboration Indicator Design: Unlike many existing systems that impose closed environments, where teachers and tutors must rely on predefined collaboration indicators, this approach allows for custom collaboration indicator design with only basic manipulation of the proposed DCIN-AGST system.

- Platform Independence: While many existing methods are platform-specific, this approach is platform-independent, allowing teachers to design and compute their own collaboration indicators on any platform without requiring expertise in computer science.

- Integrated Design and Computation: While earlier approaches primarily focus on indicator calculation, this method integrates both design and computation in a structured and automated manner.

The proposed approach makes several significant contributions to the field of e-learning analytics:

- Platform-Independent Computation: It enables the computation of collaboration indicators independently of any specific platform by applying a model transformation approach within an MDA-based process.

- Formal Model for Indicator Design: It introduces a formal model that supports the design of meaningful and adaptable collaboration indicators, aligning with teachers' observation needs.

- Automated Transformation Sequence Generation: It automates the generation of transformation sequences, allowing non-computer science teachers to design and compute collaboration indicators without requiring advanced programming skills.

By integrating these contributions, this work provides a structured, flexible, and scalable solution for collaboration indicator design and computation in e-learning environments.

REFERENCES

[1] F. Kanwal, M. Rehman, K. Bashir, and U. Qureshi, "Critical Factors of E-Learning Adoption and Acceptance in Pakistan: A Literature Review," *Engineering, Technology & Applied Science Research*, vol. 7, no. 4, pp. 1888–1893, Aug. 2017, https://doi.org/10.48084/etasr.1257.

[2] F. Henri, "Apprentissage collaboratif en mode virtuel [Collaborative Learning in Virtual Mode] (in French)," Montreal, Canada, Nov. 2002, [Online]. Available: http://www.inrp.fr/Tecne/docseminaires/fhenri021120.pdf.

[3] H. Acosta *et al.*, "Multimodal Learning Analytics for Predicting Student Collaboration Satisfaction in Collaborative Game-Based Learning," Jan. 2024, https://doi.org/10.5281/zenodo.12729802.

[4] S. Morrison-Smith and J. Ruiz, "Challenges and barriers in virtual teams: a literature review," *SN Applied Sciences*, vol. 2, no. 6, May 2020, Art. no. 1096, https://doi.org/10.1007/s42452-020-2801-5.

[5] A. Rafique *et al.*, "Integrating Learning Analytics and Collaborative Learning for Improving Student's Academic Performance," *IEEE Access*, vol. 9, pp. 167812–167826, 2021, https://doi.org/10.1109/ACCESS.2021.3135309.

[6] C. Gangloff-Ziegler, "Les freins au travail collaboratif:," *Marché et organisations*, vol. 10, no. 3, pp. 95–112, Sep. 2009, https://doi.org/10.3917/maorg.010.0095.

[7] S. A. Salloum, M. Al-Emran, K. Shaalan, and A. Tarhini, "Factors affecting the E-learning acceptance: A case study from UAE," *Education and Information Technologies*, vol. 24, no. 1, pp. 509–530, Jan. 2019, https://doi.org/10.1007/s10639-018-9786-3.

[8] M. Arnaud, "Les limites actuelles de l'apprentissage collaboratif en ligne [Current Limitations of Online Collaborative Learning] (in French)," *STICEF (Sciences et Technologies de l'Information et de la Communication pour l'Éducation et la Formation)*, vol. 10, 2003, [Online]. Available: https://hal.science/hal-00696421.

[9] A. Dimitracopoulou and E. Bruillard, "Enrichir les interfaces de forums par la visualisation d'analyses automatiques des interactions," *Sciences et Technologies de l'Information et de la Communication pour l'Éducation et la Formation*, vol. 13, no. 1, pp. 345–397, 2006, https://doi.org/10.3406/stice.2006.942.

[10] É. Gendron, "Cadre conceptuel pour l'élaboration d'indicateurs de collaboration à partir des traces d'activité [A Conceptual Framework for the Development of Collaboration Indicators from Activity Traces] (in French)," Ph.D. dissertation, Université Claude Bernard Lyon 1, France, 2010.

[11] T. Djouad, A. Mille, C. Reffay, and M. BenMohammed, "Ingénierie des indicateurs d'activités à partir de traces modélisées pour un Environnement informatique d'apprentissage humain [Engineering of Activity Indicators from Modeled Traces in a Computer Environment for Human Learning] (in French)," *Sciences et Technologies de l'Information et de la Communication pour l'Éducation et la Formation*, vol. 16, no. 1, pp. 103–139, 2009, https://doi.org/10.3406/stice.2009.994.

[12] L. S. Settouti, "Système à base de trace pour l'apprentissage humain [Trace-Based Systems for Human Learning] (in French)," in *Rencontres Jeunes Chercheurs sur les EIAH*, Evry, France, Feb. 2006, [Online]. Available: https://hal.science/hal-01513038.

[13] M. Vásquez-Bermúdez, J. Hidalgo-Larrea, F. Orozco Lara, and S. Segura Santana, "Effectiveness of Monitoring Indicators in the Architecture of a Collaborative System," in *Technologies and Innovation*, 2022, pp. 191–202, https://doi.org/10.1007/978-3-031-19961-5_14.

[14] D. Nguyen, S. Yingchareonthawornchai, V. Tekken Valapil, S. S. Kulkarni, and M. Demirbas, "Precision, recall, and sensitivity of monitoring partially synchronous distributed programs," *Distributed Computing*, vol. 34, no. 5, pp. 319–348, Oct. 2021, https://doi.org/10.1007/s00446-021-00402-w.

[15] S. Praharaj, M. Scheffel, M. Schmitz, M. Specht, and H. Drachsler, "Towards Automatic Collaboration Analytics for Group Speech Data Using Learning Analytics," *Sensors*, vol. 21, no. 9, Jan. 2021, Art. no. 3156, https://doi.org/10.3390/s21093156.

[16] *Proceedings of IEEE International Conference on Advanced Learning Technologies (ICALT 2004)*. IEEE, 2004.

[17] T. Bratitsis and A. Dimitracopoulou, "Monitoring and Analyzing Group Interactions in Asynchronous Discussions with the DIAS System," in *Groupware: Design, Implementation, and Use*, Berlin, Heidelberg, 2006, pp. 54–61, https://doi.org/10.1007/11853862_5.

[18] C. A. Collazos *et al.*, "Evaluating Collaborative Learning Processes using System-based Measurement," *Journal of Educational Technology & Society*, vol. 10, no. 3, pp. 257–274, 2007.

[19] D. Yanacón-Atía, R. Costaguta, and M. de los Angeles Menini, "Detecting collaboration skills to calculate indicators in moodle," in *Proceedings of the XVIII International Conference on Human Computer Interaction*, Jun. 2017, pp. 1–2, https://doi.org/10.1145/3123818.3123849.

[20] A. Haddouche, F. Djelil, C. Hoffmann, N. Mandran, and C. d'Ham, "Proposal of Indicators for Measuring Collaborative Writing in a Digital Learning Environment," presented at the 15th International Conference on Computer Supported Education, Apr. 2023, vol. 2, Art. no. 495, https://doi.org/10.5220/0011757700003470.

[21] C. Choquet and S. Iksal, "Modélisation et construction de traces d'utilisation d'une activité d'apprentissage : une approche langage pour la réingénierie d'un EIAH," in *Sciences et Technologies de l'Information et de la Communication pour l'Éducation et la Formation*, 2007, vol. 14, no. 1, pp. 419–456, https://doi.org/10.3406/stice.2007.968.

[22] N. Randriamalaka, "Elicitation des indicateurs pour la ré-ingénierie des scénarios pédagogiques: Approche à base de traces utilisant UTL [Elicitation of Indicators for the Re-engineering of Educational Scenarios: A Trace-Based Approach Using UTL] (in French)," in *Actes du XXVIe congrès INFORSID*, 2008, pp. 279–294.

[23] D. P. T. Ngoc, "Spécification et conception de services d'analyse de l'utilisation d'un environnement informatique pour l'apprentissage humain," Ph.D. dissertation, Université du Maine, France, 2011.

[24] I. Matazi, A. Bennane, R. Messoussi, R. Touahni, I. Oumaira, and R. Korchiyne, "Multi-Agent System Based on Fuzzy Logic for E-Learning Collaborative System," in *2018 International Symposium on Advanced Electrical and Communication Technologies (ISAECT)*, Rabat, Morocco, Nov. 2018, pp. 1–7, https://doi.org/10.1109/ISAECT.2018.8618737.

[25] A. El Mhouti, M. Erradi, and N. El Makhfi, "A Multi-Agent System of Semantic Analysis and Filtering of Modeled Traces to Calculate Interaction Indicators Favoring Collaboration in LMS," in *2019 International Conference on Intelligent Systems and Advanced Computing Sciences (ISACS)*, Taza, Morocco, Dec. 2019, pp. 1–7, https://doi.org/10.1109/ISACS48493.2019.9068907.

[26] R. Alur, "Timed Automata," in *Computer Aided Verification*, 1999, pp. 8–22, https://doi.org/10.1007/3-540-48683-6_3.

[27] D. M. Lewis, *Visualisation of Meta Object Facility Models with Human Usable Graphical Notation*. 2002.

[28] A. Dimitrakopoulou, *State of the art on Interaction and Collaboration Analysis*. 2004.

[29] G. Behrmann, A. David, and K. G. Larsen, "A Tutorial on Uppaal," in Formal Methods for the Design of Real-Time Systems: International School on Formal Methods for the Design of Computer, Communication, and Software Systems, Bertinora, Italy, September 13-18, 2004, Revised Lectures, M. Bernardo and F. Corradini, Eds. Springer, 2004, pp. 200–236.

[30] M. Reynolds, "An axiomatization of full Computation Tree Logic," *The Journal of Symbolic Logic*, vol. 66, no. 3, pp. 1011–1057, Sep. 2001, https://doi.org/10.2307/2695091.

[31] P. Bouyer, "Model-checking Timed Temporal Logics," *Electronic Notes in Theoretical Computer Science*, vol. 231, pp. 323–341, Mar. 2009, https://doi.org/10.1016/j.entcs.2009.02.044.