

**DRAFT:** Lekira A., Després C., Jacoboni P., Choquet C., Iksal S., Py D., Pham Thi Ngoc D., Using indicators during synchronous tutoring of practical work, The 11th IEEE International Conference on Advanced Learning Technologies (ICALT'11), pp. 568-572 (2011)

## Using Indicators during Synchronous Tutoring of Practical Work

Aina Lekira, Christophe Després, Pierre Jacoboni, Christophe Choquet, Sébastien Iksal,  
Dominique Py, Diem Pham Thi Ngoc,  
Université du Maine, LIUM  
Le Mans, France  
e-mail: {*firstName.lastName*}@lium.univ-lemans.fr

**Abstract—** When teachers are involved in synchronous tutoring carried out on a practical work, they need overall but semantically relevant information in order to better manage learners' activities. This information, given through indicators, enables teachers to quickly detect learners' critical situations. Thus, relying on indicators, they can intervene and choose the better remediation strategies to solve these critical situations. In this paper, we propose a way to identify, define and calculate indicators through UTL and DCL4UTL. We also conducted experimentations in order to examine the benefits of the use of indicators during practical work sessions.

*Indicators; Practical Work; Tutoring; Object-oriented Programming; Hop3x; UTL; DCL4UTL;*

### I. INTRODUCTION

The perception of learners' activities is one of the main issues in the field of Technology-Enhanced Learning (TEL). We deal with it within the framework of synchronous tutoring carried out on an activity of practical work. Tools design has been made in order to cope with this issue. Most of these tools offer teachers the opportunity to monitor learners' activities [1][2]. This simple monitoring of learners' activities is not sufficient in this context of synchronous tutoring because teachers have to fetch interesting information from a heap of other; the time they spend to follow and analyze each learner's activity can become quickly prohibitive [3].

Thus, when involved in synchronous tutoring, teachers need detailed and targeted information about learners' activities. Some attempts for that purpose have been made [4][5]. Indeed, giving teachers such relevant information allows them to quickly detect situations in which learners are in deadlock or failure. If needed, they can intervene in order to support learners in achieving their activities.

Within the framework of synchronous tutoring carried out on an activity of practical work, we identified three tasks that teachers have to manage when they are involved in the supervision of individual activities of a several learners [6]:

- Ensure that each learner performs his/her activity in the allocated time; teachers have to manage the activity progress of each learner, depending on the time.
- Support learners in their acquisition of theoretical concepts introduced in lectures or tutorials.
- Coach learners in order to gain core competencies and know-how related to the area of learning.

Our goal is to provide teachers with significant and relevant information related to each of their tasks. This information is supplied through indicators which reflect quantitatively and qualitatively learners' work [4]. From this goal, two research questions arise: how to define and calculate indicators that have to be provided in real time to teachers? What benefits bring the use of indicators?

In this paper, we deal with synchronous tutoring of several learners involved in a non-collaborative (individual) activity during practical work session. In one hand, we describe how to define and calculate relevant indicators, and in other hand, we measure benefits of the indicators use according to learners' and teachers' performance, regarding the learning domain, object-oriented programming. We conducted two experimentations by using HOP3X [7] (a TEL system), with third-year students in computer science and we compared a situation in which indicators about learners' activities are supplied to teachers to another one, in which these indicators are not supplied.

We detail the identification and calculation of indicators in the next section. HOP3X, the TEL system used during the experimentations is presented in section 3. Section 4 describes the experimentations. Experimentations results and analysis are discussed in section 5. Finally, we conclude and give an outlook.

### II. INDICATORS IDENTIFICATION AND CALCULATION

During practical work sessions, in order to support teachers when they carry out their tasks (described previously), we propose to supply them with indicators which are pedagogically interesting and calculated from learners' tracks.

An indicator is a "variable that describes 'something' related to the mode, the process or the 'quality' of the considered 'cognitive system' activity; the features or the quality of the interaction product; the mode or the quality of the collaboration, when acting in the frame of a social context, forming via the technology-based learning environment" [8]. Indicators have attributes such as their names, their values, etc.

To provide teachers with indicators in real time, first, we have to identify and define these indicators and second, they have to be calculated.

### A. Indicators Identification

If an indicator brings to teachers relevant information according to their three tasks, it has to be identified and defined from teachers' observation needs and according to the pedagogical scenario and activity objectives.

Consider an example related to task 2, in which teachers ensure learners' acquisition of theoretical concepts seen in lectures or tutorials:

*Scenario:* Create a class named Point with two real instance variables corresponding to the x and y coordinates.

*Objective:* Master the notion of class, constructor and encapsulation.

*Observation need:* Check the creation of a class named Point with two real instance variables (explicited as that in the scenario), check the creation of a Point constructor with two real parameters (implicit in the scenario), check the creation of access methods for private instance variables (implicit in the scenario).

Tab. 1 shows an example of observation need for each tutoring task and the corresponding indicators. The given example for task 2 is related to the explicit observation need described above.

TABLE I. EXAMPLES OF OBSERVATIONS NEEDS AND CORRESPONDING INDICATORS

Tutoring tasks	Observation needs	Corresponding indicators
Task 1	Observe a learner's progress compared to the rest of the group	Relative progress of a learner compared to the average progress of the group.
Task 2	Check the creation of a class named Point with two real instance variables	<ul style="list-style-type: none"> <li>- Existence of a class named Point</li> <li>- Visibility of the class</li> <li>- Number of variables in the class</li> <li>- Type of the variables in the class</li> <li>- Visibility of the variables in the class</li> </ul>
Task 3	Check the presence of comments for each method within a class	Percentage per class of methods commented by <i>JavaDoc</i> comments

An indicator reflects the gap between what learners have done and what teachers expect. We integrate this latter within the definition of an indicator through an acceptability domain of its value. The acceptability domain of an indicator value is a reference which can be a constant, a threshold, an interval or a set.

Thereby, when an indicator value is calculated during the session, to be acceptable and considered as normal, this indicator value must belong to its acceptability domain.

To provide teachers with these indicators, we use two languages we have developed: UTL (Usage Tracking Language) for indicators description and DCL4UTL for indicators calculation.

### B. Indicators Calculation

The main goal of the Usage Tracking Language [9] is to make easier the capitalization of data analysis techniques and teachers' know-how in the analysis of a learning session.

UTL is designed as a generic language to describe tracks and their semantics, including the definition of the observation needs and the means required for data

acquisition. Furthermore, it can be used to structure tracks from raw data, which are acquired from and provided by the learning system during the learning session, to indicators, which are significant for their users. These data are capitalized independently and different from any format of tracks generated by learning systems.

UTL allows describing two high-level types of data: primary data and derived data. A primary datum is not calculated with the help of other data. It could be typed as a raw datum tracked by the learning system), a content datum (as a student production for instance) or an additional datum (the pedagogical scenario, a tutor's annotation, etc.). A derived datum is calculated or inferred from primary data and/or other derived data. Intermediate data indicator types are qualified as derived data.

While UTL allows capitalizing tracks in the form of data design patterns, DCL4UTL [10] completes these data patterns by adding calculation methods for making them executable. The combination of UTL and DCL4UTL allows not only modeling tracks but also producing indicators. They are addressed to human as well as to the machine. However, both UTL and DCL4UTL are used by data analysts; teachers are not the main users of these languages. Teachers are concerned by the use of the indicators' calculation results.

Around 60 indicators were designed for 12 questions in the learning scenario. The main part of them were calculated directly in DCL4UTL, for the others it was necessary to integrate an external function which analyzed the JAVA code to deliver information such as the list of classes, of methods, their parameters, etc. The result of the external function is described as a content datum which is calculated each time it is necessary. Another important datum is the raw datum which described the event generated by HOP3X, it has the same structure but differs on the type of event and some parameters (Event CM for Manual Compilation, SQ for Select Question etc.)

Calculation of indicators depends on these learner's events such as "Run", "Compile", "Change question", etc. Each indicator is composed of a history of all values since the beginning of the session which is useful after the session for analyzing more generally learners' behavior.

We conducted two experimentations with HOP3X by using UTL. We describe in the next section this TEL system.

### III. THE TEL SYSTEM HOP3X

HOP3X is a track-based system which aids the supervision of real-time practical work in programming [7]. The tool allows teachers to monitor synchronously the activity of each learner in a group. HOP3X is composed by three applications: HOP3X-STUDENT, HOP3X-TEACHER and HOP3X-SERVER. Its architecture is presented in Fig. 1.

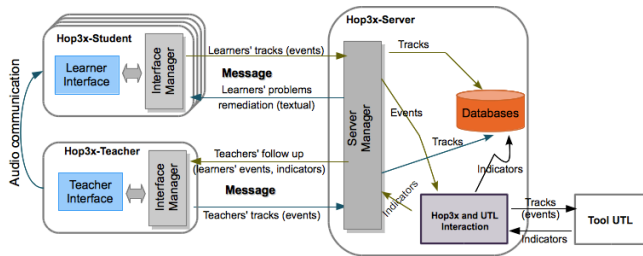


Figure 1. HOP3X general architecture.

HOP3X-STUDENT allows learners to edit, compile and run codes and programs. It also allows them to call teachers for help if needed.

HOP3X-TEACHER offers a real-time visualization of learners' activities. Through a functionality of replay, it gives teachers the possibility to see again, during session, what learners did thanks to their collected activity tracks.

A visualization interface allows teachers to monitor indicators values related to each learner's activity. Fig. 2 presents this interface.

Relying on indicators values, teachers can detect critical situations. They could then react thanks to communication tools by sending a textual message or by initiating an audio chat.

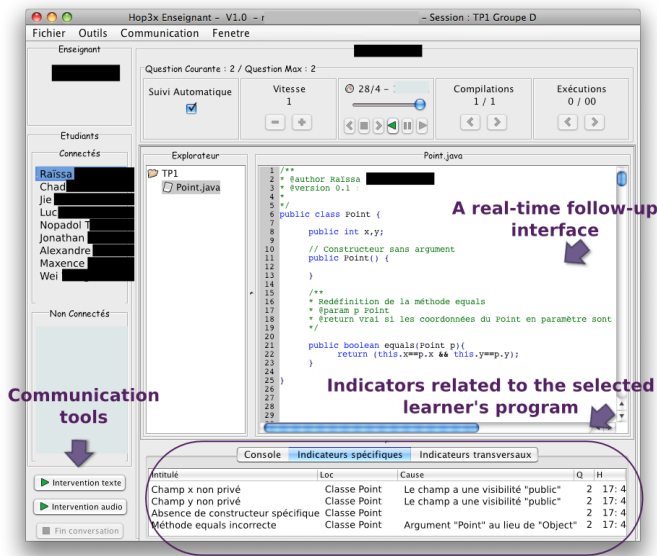


Figure 2. A monitoring interface and indicators visualization in HOP3X-TEACHER.

HOP3X-SERVER allows the collection and the storage of tracks of the learning session participants (learners and teachers). These tracks are saved as Hop3x events. For instance an event can be an add/suppression of a project or a file, a text insertion/deletion, a compilation, a run, an annotation, a textual intervention, etc. An indicator manager HUI (Hop3x and UTL Interaction) was designed to communicate with TOOLUTL which calculates indicators. To do this, this tool uses the meta-language UTL [9] and its associated language [10] DCL4UTL (c.f. section II.B).

#### IV. EXPERIMENTATIONS

To measure the benefits of indicators during learning session, in the context of synchronous tutoring, we conducted two experimentations: one with indicators available for teachers and another one without indicators.

These experimentations carried out through two college years and dealt with practical work which is provided to third-year undergraduate students of the Maine University, France. Practical work is part of a course entitled "object-oriented programming and Java". The students involved in our experimentation were neophytes in Java programming. However, they had, during the preceding term, an introductory course entitled "Introduction to Object-oriented Programming" in which they were introduced to the fundamentals of object-oriented programming such as class, object, instance, message passing, inheritance, encapsulation and polymorphism concepts.

Before each learning session, using HOP3X, in which they practiced Java programming, students attended lectures and tutorials about the notions and concepts that they would implement during practical work.

The same two teachers participated in the two experimentations and the same pedagogical scenario was used. In collaboration with the teaching team, we identified and defined 62 indicators that have been modeled with UTL.

The modality of teachers' intervention may be reactive or proactive. Indeed, teachers can intervene, either because learners directly solicit them when they are confronted with problems (reactive modality) or on their own initiative (proactive modality). Proactive interventions may be triggered by indicator values that do not match those expected. In this case, indicator values report a need for help, an interesting moment or a critical situation for a learner.

The first experimentation (**experimentation 1**) carried out from January to Mars 2009 and took place with forty eight students split into two groups of eighteen and one group of fourteen. The data collected among the three groups are similar in terms of average number of events produced per student. Thus, we gather them in **group 1**.

The second experimentation (**experimentation 2**) carried out from January to Mars 2010. It involved thirty-six students who were split into two groups of eighteen. The data collected among the two groups are similar in terms of average number of events produced per student. Thus, we gather them in **group 2**.

There was no difference between the two experimentations in terms of the amount of students' productions. On average, for a three-hour practical work, there were 4050 events per student, for group 1 and 3995 events per student, for group 2.

These experimentations allowed us to collect interactions tracks such as learners' activities, teachers' audio and textual interventions. Indicators calculated during session, were also collected.

For the analysis of these experimentations that we present in the next section, the data used were collected during session for group 2. For group 1, they were obtained

by calculating indicators thanks to tracks collected during the experimentation.

### V. EXPERIMENTATIONS RESULTS AND DISCUSSION

We want to investigate the benefits of the use of indicators during practical work. To reach this goal, we measure if the students in group 2 – in which indicators were provided to teachers – were better able to solve their problems than the students in group 1 – in which teachers had no available indicators.

For the analysis, we brought out critical situations (CS) i.e. situations in which indicators values were not acceptable. Among these situations, some have evolved positively (indicators values returned to normal at the end of the session) and others have not. For instance, declaring an instance variable without a *private* visibility is a violation of the concept of encapsulation. These situations, as seen in Tab. 1, can be detected through indicators. During a session, if this indicator value is non-acceptable, we want to know, if at the end of the session, this indicator value returned to normal or did not.

In analyzing the collected data, we observe that teachers ignored some CS. The teachers' debriefing has confirmed that they had voluntarily put these indicators aside because they realized during the session that these indicators were no longer relevant. In terms of reengineering, this finding allowed the teaching team to modify their observation needs for the scenario and to remove these unnecessary indicators from the provided indicators. Therefore, we exclude these indicators and CS in the remainder of the analysis.

The two groups are homogeneous with regard to the number of CS that appeared during the session: on average, for group 1, there was 16.85 CS per student and for group 2, there was 17.72 CS per student.

As shown in Fig. 3, 53.61% of CS were solved for group 1. For group 2, this rate is 73.52%. The final rate of CS resolution, for group 2, is 19.91 points higher than for group 1. We notice that the group in which teachers had indicators was better able to solve their CS than the one in which teachers had no indicators.

However, there is still 26.47% of non-solved CS despite teachers' interventions for group 2. To go further in our analysis, we put in perspective the CS in relation to their resolution and teachers' interventions.

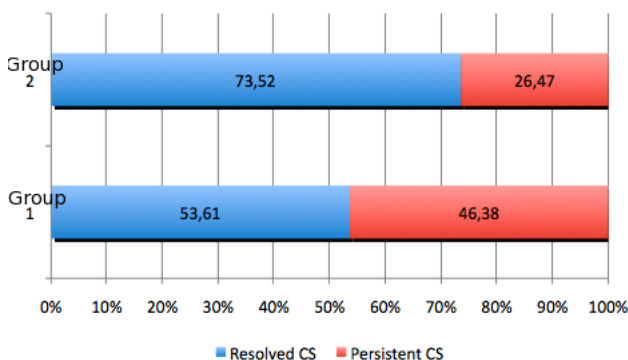


Figure 3. State at the end of session of the SC which appeared during the session.

For group 1, the technical implementation of the experimentation did not allow us to capture teachers' interventions by computer. However, by interviewing teachers, we got the number of interventions per session; it was 6 i.e. (a total of 21 interventions for the 3 groups). These interventions were few compared to the group 2 – in which indicators were provided to teachers – that had on average, 27 interventions per session (a total of 54 interventions for the 2 groups). This difference can be explained by the fact that in group 1, teachers benefited from very few information about students' activities. Therefore, they intervened mainly in response to students' requests (reactive modality). As we can see in Fig. 3, 53.61% of CS have nevertheless been solved for this group 1. We can assume that there was a high degree of self-correction: students have partially solved CS themselves.

Moreover, for group 2, we realize that thanks to indicators, teachers intervened 4.5 times more than in group 1.

To evaluate the impact of interventions in the CS resolution, Fig. 4 shows the repartition of these CS for group 2, depending on their resolutions and on teachers' interventions.

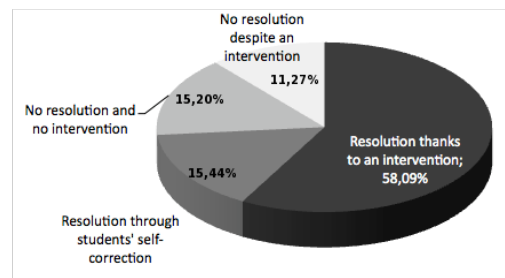


Figure 4. Repartition of appeared CS for group 2 contingent of CS resolution and teachers' interventions.

As we can see in Fig. 4, 58.09% of CS were solved thanks to teachers' interventions (it represents 83.74% of all teachers' interventions) while 15.44% were solved by students' self-correction (i.e. a rate of 73.52% of solved CS for group 2 – cf. Fig. 3).

The part of non-resolved CS despite interventions (11.27% of CS) mainly refers to problems that are difficult to solve for students. These problems deal with a key notion in object-oriented programming: the difference between overriding and overloading. In these situations, to decrease the rate of teachers' interventions failures, it would be interesting to provide teachers with a monitoring of their interventions by giving them feedback about the effects of their interventions. Thus, teachers would be able to quickly detect the interventions that have been successful and those that have failed or partially failed. In case of failure or partial failure of an intervention, teachers, alerted by the given feedback, could quickly intervene again about these non-resolved CS. In addition, giving teachers feedback about the effects of their interventions enables them to have information about their own activities, processes and actions [8]; it also allows them to be aware of the success or failure

of their interventions so as to adjust their remediation strategies and pedagogical practices (reflexive analysis).

For cases in which CS persisted at the end of the session and have not been the subject of an intervention (15.20% of CS), we remark that these CS mainly refer to two categories of problems. The first one concerns the lack of access methods for some private instance variables which are used outside the class where they are defined. The second one refers to the non-use of already created methods in the remainder of the activity. These CS appearances are not blocking for students because they can continue their activities. In such non-blocking situations, as teachers monitor several students at the same time, they have to make a choice. We can assume that they give a certain priority to CS and they decide to treat first CS that they consider as blocking. To target these situations, it would be interesting to add an indicator attribute: its critical level. This attribute would allow to provide teachers with an alert level depending on the nature of the referred situation which can be either critical, potentially critical or non-critical.

## VI. CONCLUSION AND OUTLOOK

This paper describes a way to identify and model indicators that can be calculated and provided in real time to teachers when they are involved in synchronous tutoring of practical work. Indicators description and calculation via UTL and DCL4UTL allows the capitalization of the calculation methods and their reuse.

With this paper, we have investigated the benefits of the use of significant and relevant information during real-time practical work sessions. Experimental results showed that supplying teachers with indicators about the three specific tasks they have to manage during practical work has a positive impact on students' performance; indeed, students supported by teachers who had available indicators were better able to solve their problems than the ones who were supported by teachers with no available indicators. In addition, indicators tend to improve teachers' tutoring practice. Indeed, with indicators teachers intervene much more than without: the interaction between students and teachers is enhanced. Besides, 83.74% of teachers' interventions were successful (i.e. the targeted critical situations were solved and the corresponding indicators have returned to acceptable value at the end of the session). Nevertheless, 16.25% of teachers' interventions have failed. These cases of failure relate to difficult questions. In order to decrease these cases, we propose to provide teachers with information about their activities (i.e. a feedback about the effects of their interventions). Thus, in case of failure, teachers could quickly detect this fact and react in order to

solve the critical situations. Finally, for decreasing non-solved critical situations which have not been the subject of an intervention, we propose to add an indicator attribute in order to give an alert degree to teachers: the alert degree is high when the critical situation is considered as insurmountable; that means that teachers' interventions are required.

Operationalizing these proposals is our short time objective. We also plan to supply students with some of the indicators available for teachers in order to support self-regulated learning [11]. Experimentations will be conducted in order to measure the impact of this supply on learners' performance.

## REFERENCES

- [1] D.M. Hassan, D. Leclot, B. Talon, "An online laboratory in a context of project-based pedagogy", Proc. 9<sup>th</sup> IEEE International Conference on Advanced Learning Technologies, p. 128-130, Riga, Latvia, 2009
- [2] A. Babich, K. Mavrommatis, "Virtual laboratory concept for engineering education", Proc. International Conference on Engineering and Research "Progress through Partnership", p.1043-1050, Ostrava, Czech Republic, 2004.
- [3] J-M. Labat, "EIAH : Quel retour d'informations pour le tuteur ?", Proc. Colloque International Technologies de l'Information et de la Communication dans les Enseignements d'ingénieurs et dans l'industrie, Lyon, France, 2002.
- [4] C. Després, "Synchronous tutoring in distance learning". Proc. International Conference on Artificial Intelligence, p. 271-278, Australia.
- [5] V. Gueraud, A. Lejeune, J.M Adam, M. Dubois, N. Mandran, "Supervising distant simulation-based practical work: environnement and experimentation", Proc. European Conference on Technology Enhanced Learning, p. 462-476, Nice, France, 2009.
- [6] A., Lekira, C. Després, P. Jacoboni, "Supporting the identification of teachers' intention through indicators", In 3<sup>rd</sup> International Conference on Computer Supported Education, 6-9 May, The Netherlands. (*in press*)
- [7] Hop3x: a supervision tool for practical work in programming, <http://eah.univ-lemans.fr/HOP3X/HOP3X.xml>.
- [8] ICALTS. 2004. Délivrables 1, 2, and 3 of ICALTS JEIRP: Interaction and Collaboration Analysis' supporting Teachers and Students's self-regulation, Kaléidoscope network of excellence, 2004, (<http://www.rhodes.aegean.gr/tee/kaleidoscope-icalts/>)
- [9] C. Choquet, S. Iksal, "Modeling tracks for the model driven reengineering of a TEL system" , In Journal of Interactive Learning Research, Special issue: Usage analysis in learning systems – Existing approaches and scientific issues (18)2, p. 161-184, 2007.
- [10] D. Pham Thi Ngoc, S. Iksal, C. Choquet, E. Klinger, "UTL-CL: a declarative calculation language proposal for learning tracks analysis process" . Proc. 9<sup>th</sup> IEEE International Conference on Advanced Learning Technologies, p. 681-685, Riga, Latvia, 2009.
- [11] D.L. Butler, P.H. Winne, "Feedback and self-regulated learning: a theoretical synthesis. Review of Educational Research (18), p. 245-281, 1995.