



HAL
open science

Supporting Awareness and Self-Regulation In Project-Based Learning through Personalized Dashboards

Christine Michel, Elise Lavoué, Sébastien George, Min Ji

► **To cite this version:**

Christine Michel, Elise Lavoué, Sébastien George, Min Ji. Supporting Awareness and Self-Regulation In Project-Based Learning through Personalized Dashboards. International Journal of Technology Enhanced Learning, 2017, 9 (2/3), pp.204-226. 10.1504/IJTEL.2017.084500 . hal-01384211

HAL Id: hal-01384211

<https://hal.science/hal-01384211>

Submitted on 31 May 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Supporting awareness and self-regulation in project-based learning through personalised dashboards

Christine Michel*

INSA-Lyon, Univ Lyon,
LIRIS, UMR 5205, CNRS,
Villeurbanne F-69621, France
Email: christine.michel@insa-lyon.fr
*Corresponding author

Elise Lavoué

IAE Lyon, Université Jean Moulin Lyon 3,
LIRIS, UMR5205, CNRS,
Lyon F-69003, France
Email: elise.lavoue@univ-lyon3.fr

Sébastien George

UBL, Université du Maine,
EA 4023, LIUM,
Le Mans F-72085, France
Email: sebastien.george@univ-lemans.fr

Min Ji

INSA-Lyon, Univ Lyon,
LIRIS, UMR 5205, CNRS,
Villeurbanne F-69621, France
Email: jimin@sspu.edu.cn

Abstract: Project-based learning (PBL) enables learners to carry out challenging and authentic projects through investigations. However, PBL is difficult to implement successfully because learners often lack the self-regulation skills required to monitor, reflect, manage and assess their project activities and learning. Furthermore, most learning systems rarely offer possibilities to monitor and reflect on their project and learning processes. Hence, in this paper, we propose a general architecture of project-based learning management system. It enhances learners' reflective processes by supporting them in creating customisable indicators by exploring their traces. We developed an implementation named DDART and conducted an experiment in order to evaluate its usability and perceived utility. We found that this system supports learners to reflect and regulate their activities and learning, even if the indicator creation could be difficult for the novices.

Keywords: activity trace; dashboard; project-based learning; project-based learning management system; reporting trace; self-regulated learning.

Reference to this paper should be made as follows: Michel, C., Lavoué, E., George, S. and Ji, M. (2017) 'Supporting awareness and self-regulation in project-based learning through personalised dashboards', *Int. J. Technology Enhanced Learning*, Vol. 9, Nos. 2/3, pp.204–226.

Biographical notes: Christine Michel is an Associate Professor, habilitated to supervise research in the Department of Industrial Engineering, INSA Lyon. She is a member of the LIRIS Research Laboratory. She received her PhD from the University Claude Bernard Lyon 1 in France in 1999 and then joined the University of Bordeaux III to work on the analysis of technology-enhanced learning (TEL) system uses with log traces. In 2003, she joined INSA Lyon and focused her work on the design of mixed method, combining qualitative and quantitative data, to observe learning and teaching activities. These methods are used to analyse TEL adoption and appropriation and to design TEL systems using traces of activity for reflexive practice.

Elise Lavoué is an Associate Professor of Computer Science at the University Jean Moulin Lyon 3 (France), LIRIS lab. Her research interests include the design of learning environments to support self-regulated learning and learners' engagement by using interaction traces, in the fields of technology enhanced learning and computer-supported collaborative learning (CSCL). She has authored and co-authored more than 60 publications, including journal articles, book chapters and conference papers in these areas. She served as organising or program committee member for the EC-TEL, CSCL, EMOOCs, LAK and ICWL international conferences.

Sébastien George is a Full Professor of Computer Science since 2013 at University of Maine, France. He teaches at the Computer Science department of the Institute of Technology of Laval. He is the head of the team IEIAH (working in the field of technology enhanced learning) at LIUM Laboratory. He received his doctoral thesis from the University of Maine in France in 2001. Then he did a postdoctoral fellowship at the TeleUniversity of Quebec in Canada. He joined INSA Lyon in 2002. He has been qualified as HDR (Habilitation thesis) in 2010, an accreditation to supervise PhD students. His major fields of interest are Human-Computer Interactions, CSCL, computer-mediated communication, assistance to human tutoring in distance education and serious gaming.

Min Ji is a Lecturer in the School of Intelligent Manufacturing and Control Engineering, Shanghai Polytechnic University. She works in the information management and information system laboratory. She received her PhD degree from Institut National des Sciences Appliquées de Lyon in 2015. Her researches focus on the methods to support learners' self-regulation in project-based learning by using traces, as well as the design and construction of management information system.

1 Introduction

Project-based learning (PBL) can be defined as “a teaching method in which students gain knowledge and skills by working for an extended period of time to investigate and respond to a complex question, problem, or challenge” (Buck Institute for Education, 2014). In such educational settings, learners need to plan their learning, organise their work, implement several learning strategies and use different tools. In a recent study, English and Kitsantas (2013) show that PBL facilitates knowledge acquisition and retention, supplies learners with the opportunity to improve their skills in problem-solving, investigative activities, decision making, analysing and evaluating information, thinking critically, working cooperatively and communicating effectively. However, they also show that learners are poorly self-directed and lack of self-regulation skills required in PBL. Therefore, it is important to support self-regulation processes for learners involved in PBL activities to increase the success ratio (Chang, 2007).

Self-regulated learning (SRL) was defined by Zimmerman (1989) as ‘the degree to which learners are metacognitively, motivationally and behaviourally active participants in their own learning’. Self-regulated learners are able to set their learning goals, to plan, to conduct and then to regulate and to evaluate the learning processes independently (Narciss, Proske and Koerndle, 2007). One way to help learners to become self-regulated is to provide them with indicators on their progress, thoughts and feelings during the PBL activity (Michel, Lavoué and Pietrac, 2012). They can so reflect on their way to carry out the activity, re-evaluate their strategies and plan new ones.

Indicators can usually be visualised at a glance on dashboards (Lamprey and Fayek, 2012). However, we observe that dashboards are often underused by learners when carrying out project activities. One reason is that most of the dashboards offer predefined indicators and learners may not know how to use them in a proactive approach (Littlejohn, Margaryan and Milligan, 2009). One way to engage learners in metacognition processes supported by dashboards is to involve them in the construction of their own indicators. In this way, they are led to reflect on their project both when building their indicators and when visualising them.

Another limitation of existing dashboards is that they usually focus on a particular type of information, like learners’ participation (Janssen, Erkens and Kirschner, 2011), behaviours (Kimmerle and Cress, 2008) or knowledge (Sangin et al., 2011). However, supporting SRL implies considering all the following dimensions: cognitive, motivational and behavioural (Pintrich, 2004). This limitation can be explained by the fact that most of the systems are based on the interaction traces with the learning environment and so acquire only information on the users’ activities with the system. New ways have to be explored to also integrate information on users’ thoughts and feelings during PBL.

Our approach is based on the assumption that providing learners with a personalised dashboard that present information on the cognitive, emotional and behavioural dimensions of SRL will help them to manage their projects. In this paper, we propose a general architecture of project-based learning management system (PBLMS) that allows learners to build their own indicators, by defining the information, the calculation and the visualisation. The architecture combines two types of traces: activity traces that correspond to users’ actions recorded directly by the system during the learning activities; and reporting traces that are reported by the learners themselves to explain how they (or their peers’) have carried out their activities out of the learning environment. The reporting traces are integrated with the interaction traces to produce rich indicators on

learners' progress, thoughts and feelings. We also present an implementation of this architecture with the DDART system that offers a user-friendly interface for learners to create personalised indicators. We report the results of a study we conducted to evaluate the usability of this system.

In the following sections, we first study the types of traces used to produce indicators and existing dashboards proposed in the literature. We then detail the architecture of PBLMS and its implementation with the DDART platform. We finally evaluate the possibilities offered by the features of DDART for the creation of some existing indicators and discuss the limitations of this system. We conclude with the perspectives offered by this research and describe our future work.

2 State of the art

In this section, we first study the types of traces used in technology-enhanced learning (TEL) applications to produce indicators. We then analyse the dashboards proposed to students to monitor and reflect on their learning processes.

2.1 Traces and indicators in TEL

Most of the tools that produce indicators use the users' activity traces on the learning environment and are designed for tutors. For instance, ESSAIM (Després and Leroux, 2002) presents information on the learners' activity progresses from a global view. FORMID (Gueraud and Cagnat, 2004) has been developed to help tutors to monitor learners' synchronous individual and group project activities. Some tools offer indicators to support learners' monitoring and reflection processes. For instance, CourseVis (Mazza and Dimitrova, 2007) uses activity traces produced by WebCT in order to compute graphical indicators about the learning behaviours, social characteristics and cognitive evolutions of distance students. TrAVis (May, George and Prévôt, 2011) is a reflective tool giving information to learners on the way they carry out discussions or other collaborative activities. All these systems offer indicators based on interaction logs collected automatically by the learning system. Because learners cannot access to the trace collection process, they have no possibility to complete the traces by other information (such as their goals, thoughts, emotions and judgements). Furthermore, learners also use other computer software (Skype, YouTube, Firefox, MS office) and carry out outdoor activities (visit factories, face-to-face discuss, collect temples for testing). Unfortunately, these data are not collected and analysed by the systems.

Blogging and electronic portfolios provide learners with a place to record manually the information related to their learning and support them to adjust their learning processes through reflection. For instance, EnquiryBlogger (Ferguson, Buckingham Shum and Deakin Crick, 2011), a WordPress plugin, generates three indicators providing visual feedback to learners on their progress and affective state based on their categorisation of the blog contents. Other researchers (Glaser-Zikuda et al., 2011) indicated that the use of electronic portfolio helps learners to set goals and plan, monitor and reflect on their learning, and assess their performances in comparison with the goals. However, it is difficult for learners to focus on subjects related to the learning activities as they can write anything they want freely. Moreover, there is no guide helping them to organise the content to be more effective to support reflection and learning processes.

Finally, most blogs and e-portfolios are non-structured texts and the contents are hard to be analysed automatically by the e-learning systems. Only the Pco-Vision system dedicated to PBL (Michel, Lavoué and Pietrac, 2012) provides learners with a structured self-report tool, but learners can only answer to predefined sentences and cannot write their own sentences.

Some systems are based on hybrid traces (activity and reporting traces) to produce indicators. For instance, The Learning Kit Project (Winne, Hadwin and Gress, 2010) is a collaborative learning platform that proposes different communication and collaborative writing tools (gStudy, the Coach, gChat and the LogAnalyzer) to support distance learning. Activity traces (analysed by LogAnalyzer) and reporting traces (written in gChat) are used to support self-regulation in learning. However, the system does not analyse the contents of the chats. Furthermore, activity traces and reporting traces are not really integrated together. The 'MIRROR Integrated User Profile' application (MUP App) (Fessler, Wesiak and Luzhnic, 2014) is developed to integrate, synthesise, analyse and visualise traces captured by several different applications in order to arouse and support higher level reflection possibilities in particular with two applications: KnowSelf and MoodMap. The authors found that combining data collected from different applications, analysing and visualising them together can further promote reflective learning and enhance awareness of the work life (Fessler, Wesiak and Luzhnic, 2014). Nevertheless, the activity traces and reporting traces are not integrated together and some information are ignored, like the judgements and activities conducted outside the environment.

Considering the researches presented in this section, we observe that there is no system that manages the integration of both reporting and activity traces by considering all the types of information (cognitive, affective, behavioural, contextual) that can be reported by the learners during PBL activities. Indicators are based either on activity traces or reporting traces (in blogs and ePortfolio) and focus on a specific kind of information on the project. The MIRROR Integrated User Profile is the initiative more close to our approach but the traces come from separate applications and are not integrated to produce indicators. We propose in Section 3 an approach to integrate the two types of traces in order to generate rich indicators.

2.2 Personalised dashboards for students

A dashboard used in PBL should present information about the project goals and activities at a glance, on the shape of indicators that allow easy navigations to more complete information on analysis views (Michel and Lavoué, 2011). Most of the dashboards proposed in the literature are composed of predefined indicators that cannot be modified by the learners. For instance, Study desk (Narciss, Proske and Koerndle, 2007) presents the percentage of correctly solved tasks, partly correctly solved tasks, incorrectly solved tasks and further tasks in a bar chart, and give an evaluation of the overall performance. From these indicators, learners can monitor the progress of their tasks. Mastery Grids (Loboda et al., 2014) is a learning platform that helps learners to be aware of their way of learning and where to allocate their efforts next. The activity traces are presented into predefined grids and learners can know their own progresses, the group progresses and the differences on a specific activity.

Some dashboards offer possibilities to users to define personalised indicators. We identify three levels of personalisation: the low level supports users in setting some

simple parameters to filter the results of the indicator, the medium level allows users to define the calculation functions and to change the visualisations of the indicator, the high level enables learners to manipulate the traces used in the indicator (add some information into traces and select traces to be calculated). The upper levels contain the lower levels. LARAE (Charleer et al., 2014) is an example of dashboard that offers a low level of personalisation. It provides the overview, the context and the content of learners' posting, commenting, twittering and re-tweeting activities. Users can personalise the indicators by filtering the information according to the user and to the module. Govaerts et al. (2012) proposed the SAM application that presents visualisations of learners' actions from the data tracked in the learning environments. Learners are also provided with some filters on the information to present on the visualisations.

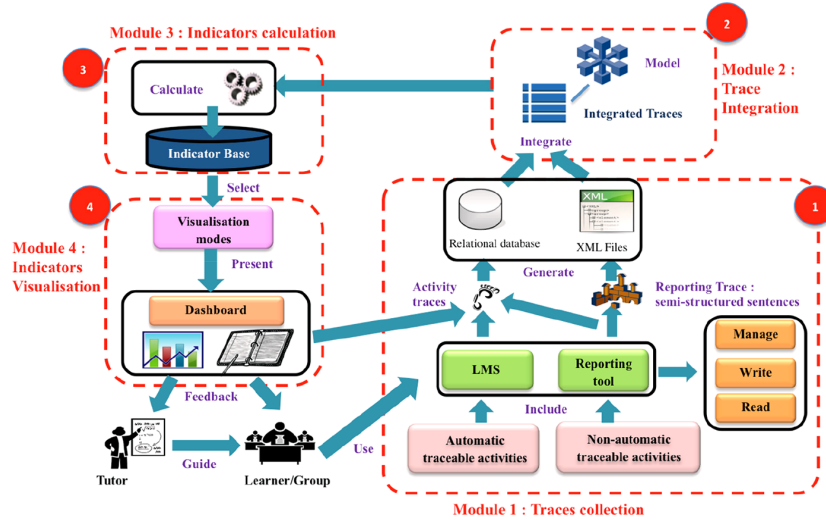
Only a few dashboards offer a medium level of personalisation. The academic analytics tool (AAT) (Graf et al., 2011) allows tutors and researchers to analyse learners' behaviour data. It explores the activity traces recorded by the system. The AAT allows users to extract specific information from the activity traces they are interested in and to select the analysis methods they want to perform through a SQL query GUI. Tutors can not only apply a set of predefined analysis methods (count, sum up) but also create custom ones. However, the only visualisation mode of the results is table, and specific computer skills are needed to write SQL queries. The GINDIC (Gendron, 2010) system can also allow researchers to create and manage personalised indicators. They need to define the indicators and to implement the calculation on the traces in order to get the indicator results. Users can assign different visualisation modes to a particular indicator, but it requires them to have a computer background because many parameters need to be set (for example, selecting rules, editing calculation operators, assigning the rights).

In conclusion, the higher the personalisation level is, the higher the computer background requirement is. Furthermore, there is no dynamic dashboard proposed to learners with high-level personalisation functions that allow them to choose the traces, the calculation mode and the visualisation associated to their indicators. In the following section, we present the general architecture of a system that allows users to create indicators based on reporting and activity traces and the implementation of this process into a platform called DDART.

3 General architecture of a project-based learning management system

We propose a general architecture of a PBLMS that can improve learners' self-regulation skills during PBL. We define a PBLMS as "a computer system that provides participants with some management tools to support and monitor the learning activities in order to solve a complex issue, problem or challenge". This system supports learners in the creation of personalised indicators related to their activities by collecting both activity and reporting traces. Learners can build their own dashboards by using these traces. Figure 1 presents the global architecture of a PBLMS. Considering that existing LMS still provides a variety of collaborative tools (e.g. communication, knowledge sharing, document downloading/uploading) to support a range of activities, we decided to take advantage of these tools in PBLMS. Four modules support the data treatment process: the collection of activity and reporting traces, the integration of traces, the calculation of indicators and the visualisation of indicators.

Figure 1 General architecture of a project-based learning management system (see online version for colours)



3.1 Trace collection

At the first step of the PBLMS process, two types of traces are collected: activity traces and reporting traces. The activity traces correspond to the interactions of the learners with the PBLMS, including the interactions among learners supported by the PBLMS. They are automatically captured and stored in a relational database. The reporting traces come from a reporting tool we have developed (Ji et al., 2013).

Projects require learners to carry out different types of activities, some of them outside the learning environment. The objectives of the reporting tool are:

- 1 to help learners to reflect on their project activities and learning
- 2 to collect information on the activities which cannot be traced automatically by the system, such as the project plans, the project activities carried out without PBLMS, the peer-judgements, learners' states of minds and so on.

It supplies three functions: write a new report, manage one's own reports and read the other members' reports.

There are two kinds of reports: goal reports and activity reports. Goal reports are used to plan the activities and to support the forethought phase of SRL. Most of the goal reports are written at the beginning of the project, but they can be modified during the project if needed. Learners can produce activity reports regularly along the project. Activity reports are used to store reporting traces on the tasks carried out, the judgements about the work progress, the acquisition of knowledge and skills, and also the state of mind.

We propose that the learners are guided in the writing of the reports by the way of semi-structured sentence models. Examples of these models are: 'The project goal is to do what from when to when', 'My skill/knowledge should reach the level of very good/good/neutral/not very good/not at all good', 'I did what with whom when where'.

The writing of these reports supports learners' self-reflection on all the dimensions of SRL (cognition, motivational and behavioural). The two types of reports allow the comparison between the goals set by the learners and the information reported by the learners during the project.

3.2 Trace integration

The trace integration consists in integrating activity traces and reporting traces together based on a uniform integrated trace model. The traces are stored in the form of *Integrated Traces (IT)* according to the following model:

$$IT = \{Id, A, L, Ca, Co, To, P, RO, BT, ET, C\}$$

With:

- Id: the trace id.
- A: the activity carried out by learners during the project, the subject of the plan or judgement, e.g. viewing forum, organising a brainstorm or the skill assessed by the learner.
- L: the learner who carried out the activity. It can be any member of the project group.
- Ca: the category of the activity: planning, project tasks and judgement.
- Co: the content of the activity. It includes the detailed contents produced during the activity, e.g. the content of forum posts or chat messages sent by a learner, the value of a judgement, the topic of a meeting, the steps of a plan.
- To: the tools used by learners to accomplish the activity, e.g. wiki, chat, forum, Dropbox, Firefox, NetBeans.
- P: the place where learners accomplished the activity, e.g. home, university or a virtual place.
- RO: the related object of the activity.
- BT and ET: the begin time and the end time of the activity.
- C: the comments on the traces.

For example, when the user Tom looks at the forum post on Moodle to know the project task he has to carry out the 1st April at 8am, the integrated trace produced is $IT1 = \{T210, View\ forum\ post, Tom, Project\ task, -, Forum, Moodle, T132, 01-04-2015\ 08:00, 01-04-2015\ 08:10, -\}$. When Tom declares on the reporting tool that he has read, at home with Mary, a pdf academic paper about how to do a project requirements analysis the 1st April from 9 am to 9:30 am, the integrated trace produced is $IT2 = \{T211, Read\ an\ academic\ paper, Tom, Project\ task, How\ to\ do\ project\ requirements\ analysis, Adobe\ reader, Home, Mary, 01-04-2015\ 09:00, 01-04-2015\ 09:30, I\ read\ an\ academic\ paper\ about\ how\ to\ do\ project\ requirements\ analysis\ with\ Mary\}$. When Tom declares on the reporting tool that he has a low opinion about the investment of Julien in the project after looking at the document he puts on the dropbox, the integrated trace is $IT3 = \{T213, Julien, Tom,$

Judgment, Low, dropbox, -, T351, 01-05-2015 09:00, 01-05-2015 09:00, The document produced by Julien is very poor}.

This IT model has been defined according to two requirements:

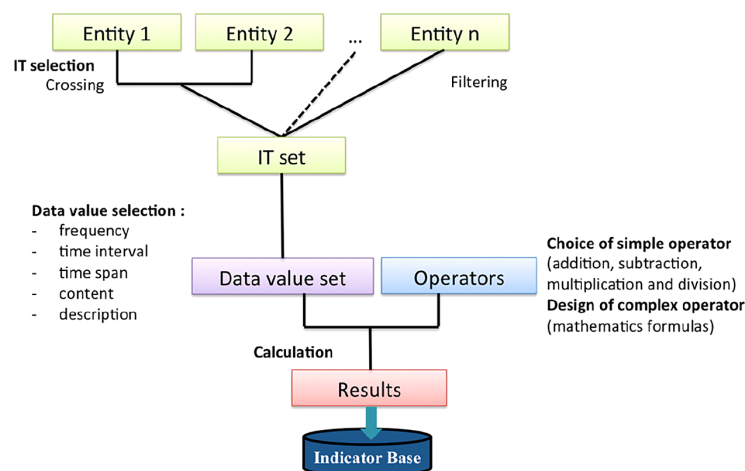
- 1 to support the collection and integration of both activity and reporting traces
- 2 to support the acquisition and production of knowledge on all dimensions of SRL.

All the attributes of the proposed IT model are adapted to both the activity and reporting traces, except C that is only dedicated to reporting traces (bringing contextual information on the trace). In the examples, IT1 is an activity trace, whereas IT2 and IT3 are reporting traces. The attributes also are generic enough to be used to manage all types of data linked to the SRL process. In the examples, IT1 and IT2 concern the behavioural dimension, whereas IT3 is about the cognitive dimension.

3.3 Indicator calculation

The calculation process of the indicators can be divided into four steps (Figure 2): IT selection, data-type selection, operator selection and calculation.

Figure 2 Indicator calculation process of PBLMS (see online version for colours)



- The IT selection corresponds to the choice of the entities the learner wants to visualise in the indicator. An entity is a value given to an attribute of the IT model presented in the previous part: A* (A, Ca, RO), L, To, P and Ti (time interval between BT and ET). Crossing (with a Cartesian product) and filtering consist in keeping only the interesting IT with specific values (e.g. only two members of the group, or a specific place where an activity has been carried out).
- The data-type selection consists in choosing the type of data used for the calculation from the IT set. Five types of data are proposed: frequency, time interval, time span, content and description. The content and the description refer, respectively, to the attributes Co and C from the IT model. The frequency, the time interval and the time spent are calculated with the values of the selected IT set.

- The calculation operators are mathematical formulas. They can involve simple operators (addition, subtraction, multiplication and division) or complex ones, designed by the users themselves.
- The calculation step produces the result by applying the operators on the values of the selected set of data.

For example, in order to build a meaningful indicator for Pascal showing the frequencies and the context of use of the chat and the NetBean tools from date D1 to date D2, the IT selection has to be 'Time' for Entity 1 with a filter 'from D1 to D2' and 'tools' for Entity 2 with a filter 'chat, and NetBean'. A last entity has to be chosen to select only the data linked to Pascal's activities: Entity 3 corresponds to 'Learner' with a filter 'Pascal'. The two entities crossed are Entities 1 and 2. The data types of the crossing results are 'Frequency' and 'Description'. In order to have also an estimation of the average frequency of each tool, a complex operator 'AVG' must be designed to calculate the average frequency by tool. The results will be the frequency of use of each tool for each date existing in the integrated trace (IT) selected, the average use for each tool during the whole period from D1 to D2 and the comment associated with each IT selected.

3.4 Indicator visualisation

The indicator visualisation module presents the results of the indicator calculation step into visually understandable diagrams and charts. It aims at synthesising complex information and so reducing the learners' cognitive load. The process is divided into three steps: the selection of the visualisation, the adjustment of the data format and the indicator presentation.

The selection of the visualisation is performed by the users. Providing learners with several possible visualisations allow them choosing the most relevant according to their preferences. Learners can also observe the indicator results from different dimensions. For example, line charts enable to observe the trends based on the time sequences, pie charts allow comparing the proportions of different elements and scatter charts help to discover the different clusters.

The adjustment of the data format is done by the system automatically according to the data value selection: *table*, *line*, *pie*, *histogram diagrams* are adapted for frequency and time interval data, *Gantt diagram* for time span data and *table* or *text* for content or description data. Considering that each visualisation mode has different data format requirements, it is necessary to preprocess the calculation results according to these requirements. If the selected visualisation is not suitable for the calculation results (for example, the calculation results are time spans of several project activities, but the selected visualisation mode is a bar chart and not timeline chart), a message is sent to the user to inform that the system fails in adjusting the data format. After the calculation results are preprocessed, they are sent to the corresponding visualisation algorithms and the indicator results are presented into the chosen diagrams.

3.5 Discussion on PBLMS to support SRL

Zimmerman's (2002) model defines three cyclical steps of SRL: forethought, performance and self-reflection. The forethought phase consists in analysing the task, setting goals and establishing strategies to attain them. The performance phase occurs during the task and refers to the monitoring and the control of the task. The self-reflection phase occurs after the task and consists in judging the task and reacting according to this judgement (positively and negatively). The PBLMS architecture was designed to support these three steps of the SRL processes during PBL activities.

The use of the reporting tool mainly supports the forethought and self-reflection phases. Building the goal report encourages learners to set goals (Michel, Lavoué and Pietrac, 2012). Building the activity reports favours the reflection on the way they carried out the tasks, their state of mind or mood and other information on the task. We can also notice that the goal and activity reports can concern the individual learner, the group or the project. This implies that the regulation processes can occur at several levels identified in (Järvela and Hadwin, 2013; Lajoie and Lu, 2011): individual, dyadic and group. For instance, when students build a goal and activity report for the group, they can compare their own goals and progress to those of the others.

The calculation of the indicators also supports the forethought and self-reflection phases. By designing the indicators, learners reflect on their goals and the way they can verify they have achieved them. By manipulating the activity and reporting traces when designing the indicators, learners are led to reflect on them. Moreover, allowing learners to build their own indicators (including the choice of the visualisation) engages them not only in the PBL activity but also in the monitoring of this activity.

Finally, the visualisation of the built indicators supports the performance phase. In fact, learners can use the indicators to monitor the project and to judge their progress according to the planned goals. They can also be aware of the judgements and tasks reported by the other members of the group. For example, learners can build an indicator on the knowledge level acquired by a learner or the group as a whole. This kind of indicator should present the current levels of the knowledge and skills that learners have acquired (according to the activity reports), in comparison with the targeted levels defined in the goal reports. For instance, it can be done in the PBLMS by

- 1 crossing Ca(planning), Ca(judgement), A(developing in PHP) and A(manage a project) with Ti (time period)
- 2 filtering on L(learner name)
- 3 choosing data value (content, description)
- 4 selecting visualisation (bar, table).

The collection and representation of subjective (reporting traces) and objective traces allow learners being aware of the differences between their own judgement, and the traces automatically collected by the system. Moreover, the visualisation of comments (C) and produced contents (Co) facilitates the interpretation of indicators and decision-making. This kind of indicator participates in the self-monitoring and self-judgement processes.

Another added value of the PBLMS is to be generic and very open. Indeed, the IT model includes the five key focus of observation: what (A, Ca, Ro), who (L), how (To), where (P), when (Ti). It allows taking into account various aspects of a person, a project and a group management. Various indicators can be built, like the time spent on tasks per students, the quality of the work or the learner's mood. These indicators can concern the individual and/or the group, in the three dimensions of SRL (cognitive, motivational and behavioural). By using the PBLMS, learners can so develop the self-regulated skills required in PBL.

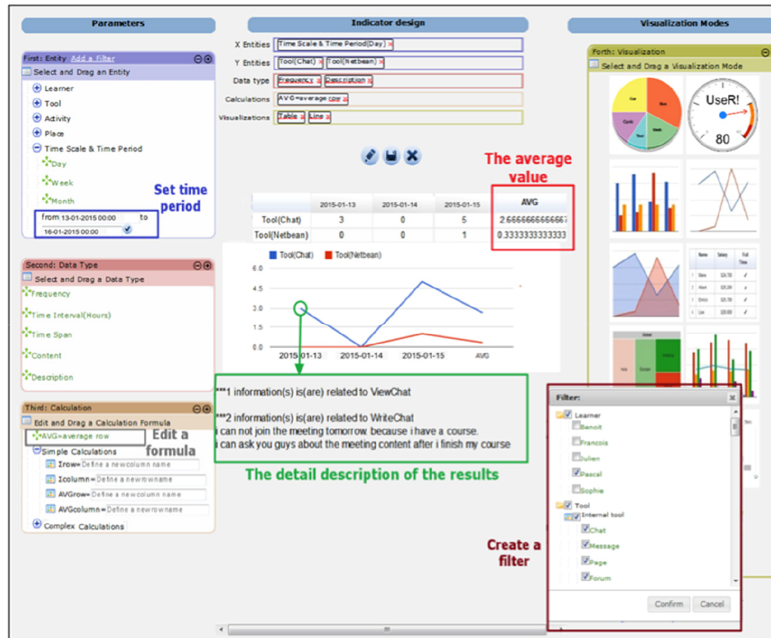
4 DDART: a customisable dashboard to create personalised indicators

We developed an implementation of PBLMS based on the proposed general architecture, which integrates the learning system Moodle. Moodle is one of the most popular open-source e-learning platforms and offers a wide variety of tools to support the teaching and learning processes. The implementation is composed of Moodle and DDART (Dynamic Dashboard based on Activity and Reporting Traces). DDART includes a reporting tool and a dynamic dashboard, which are developed to support reflection and monitoring functions. The current article only focuses on the creation of personalised indicators and dynamic dashboards. The readers interested in more complete information on the reporting tool may consult the paper (Ji et al., 2013).

4.1 Interface to create personalised indicators

Figure 3 illustrates the interface of the dynamic dashboard to create a new indicator. This interface is composed of four parts:

- 1 The 'parameters' part, on the left side, contains the list of all the entity values, the data types and the calculation operators which are available for creating an indicator, so that learners can view all the parameters intuitively.
- 2 The 'indicator design' part, in the up centre, allows learners to place the parameters by dragging and dropping from the 'parameters' part, which supports learners to set or modify the parameters easily.
- 3 The 'visualisation modes' part, on the right side, supplies eleven presentation modes the learners could choose, which allows learners to display the results from different points of views.
- 4 The 'result' part, in the bottom centre, presents the indicator results.

Figure 3 Interface of DDART to create an indicator (see online version for colours)

When the parameters are set at once, the results can be visualised and the learners can manage indicators by saving, cancelling or modifying them.

An example of indicator creation is given in Figure 3. This indicator is built to observe the frequencies of use of the 'Chat' tool and the 'NetBean' tool used by Pascal from 13-01-2015 0:00:00 to 16-01-2015 0:00:00. We use this example to describe an overview of the entire process for building an indicator. In the 'Indicator design' part, the 'X Entities' pane and 'Y Entities' pane are used to drop the entity values selected from the 'Entity' block. The values placed in these two panes are crossed together. In this example, the 'Time scale & time period (day)' entity value is listed in the 'X Entities' pane and the 'Tool (Chat)' and 'Tool (NetBean)' entity values are dropped in the 'Y Entities' pane. The time period is set in the 'Entity' block (highlighted with the left top frame in Figure 3).

The filter message box helps learners to filter the traces, if they want to narrow the data further. In Figure 3, the indicator is related to the information of the learner 'Pascal'. Hence, a filter is necessary to exclude the traces that are not related to Pascal from the database by unchecking other members' names (highlighted with the right bottom frame in Figure 3).

Learners can drag a data type from 'Data type' block and drop it in the 'Data type' pane to calculate the indicator values. In Figure 3, the indicator is created to calculate the usage frequency of each tool used by Pascal every day, so the data type of 'Frequency' can be dragged and dropped in the 'Data type' pane. If learners want to calculate the time spent on each tool every day, they can drag the data type of 'Time interval'; if they want to calculate the usage time span on each tool every day, they can drag the data type of 'Time Span' and if they want to know the interaction contents of each tool every day (chat content, wiki content ...), they can drag the data type of 'Content'. In this example,

the data type of 'Description' is also selected. When the mouse is over the results, explanations on indicator results are provided (highlighted with the middle bottom frame in Figure 3).

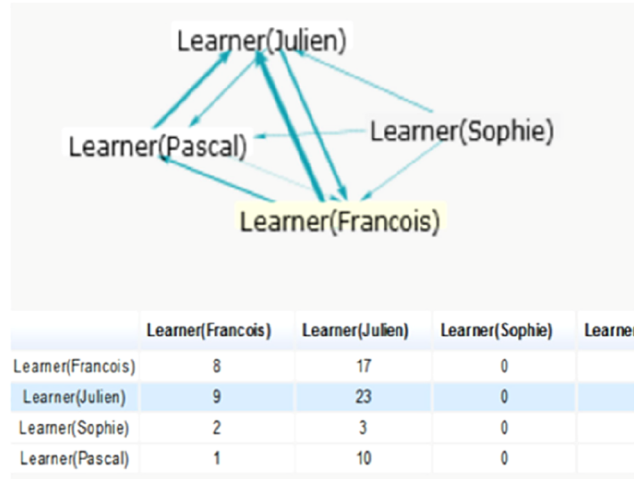
The 'Calculations' pane is used to place the mathematics formulas edited by learners in the 'Calculation' block. In the above example, in order to calculate the average frequency of each tool, learners need to edit a formula ($AVG = \text{average row}$) in the 'Calculation' block and to drag it to the 'Calculations' pane (highlighted with the left bottom frame in Figure 3). The average values can be calculated and presented (highlighted with top right frame in Figure 3).

Learners can select one or several modes to visualise the indicator results from different points of views. They can drag any appropriate visualisation modes and drop them in the 'Visualisations' pane. DDART supplies eleven visualisation modes: *Table*, *Gauge chart*, *Pie chart*, *Bar chart*, *Gantt chart*, *Combo chart*, *Line chart*, *Area chart*, *Scatter plot*, *Tree map chart* and *Network chart*. We import the Google Visualisation API to generate the first ten visualisation modes and import Springy, which is a force-directed graph layout algorithm, to produce the network chart. In the example shown in Figure 3, the visualisations of line chart and table are selected. Hence the indicator is presented in these two forms.

4.2 Illustration of indicator creation

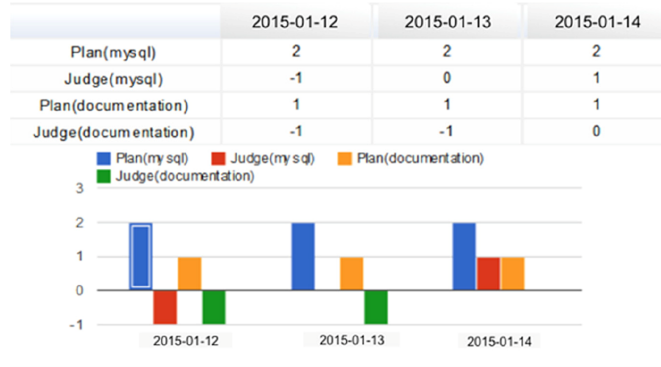
In order to test the ability of DDART to produce the most common indicators, we create the main indicators proposed in existing TEL works. We use simulated data to create these indicators. We suppose there is a small group involved in a project. The aim of the project is to develop a small website. This group is composed of five members: Sophie, Julien, Pascal, Benoit and François. The project begins from 11th Jan, 2015 to 16th Jan, 2015. The group members have already done some project tasks (e.g. communicating, sharing information, discussing) in the Moodle environment and reported their activities outside of Moodle and also their reflections and assessments. Next, we use all the traces produced in these 6 days as the data source to recreate the existing indicators with the help of DDART. As an example, we describe below how we produce two specific indicators.

The literature often refers to indicators reflecting social relationships between users. For instance, Reffay and Chanier (2003) proposed a forum indicator represented as a directed and valued graph $Gf(A,M,P)$. A is the set of agents, M is a matrix $A \times A$ in which the value of each couple (a,b) in $A \times A$ represents their interaction frequency. Thus, a network presents the interaction between learners. Figure 4 is the interface of DDART to create this particular indicator. Hence, the values of the 'Learner' entity are 'Learner (Francois)', 'Learner (Sophie)', 'Learner (Julien)' and 'Learner (Pascal)' which are listed in the 'X Entities' pane and 'Y Entities' pane. The visualisation modes are 'Social network' and 'Table'. We have to set a filter in order to focus on the interactions of viewing and writing forum posts. The 'Frequency' data type can calculate the interaction frequencies between two learners. The indicator results are presented in the 'Result' part (Figure 4). From this indicator, we can know that Francois, Julien and Pascal connected frequently while nobody contacted Sophie.

Figure 4 Creation of a ‘forum graph’ indicator (see online version for colours)

Another indicator creation could illustrate a knowledge level of a learner or a group. For instance, Michel, Lavoué and Pietrac (2012) proposed an indicator that can present the current levels of the knowledge that learners have acquired, in comparison with their target levels defined in the project plans. Figure 5 presents the interface to create this ‘knowledge development’ indicator with DDART. In the reporting tool, learners can set their knowledge target levels. During the project, they can assess and record their knowledge levels in the activity reports at any time. The reporting tool provides a five-point Likert scales (from ‘very good’ to ‘not at all good’, corresponding to the numbers from 2 to –2) to help learners to assess themselves. We suppose this indicator focuses on two knowledge of Benoit: MySQL and documentation. In the ‘Indicator design’ part, the ‘Time Scale & Time Period (Day)’ entity value is dropped in the ‘X Entities’ pane so that the knowledge development can be presented according to the date. The time period is set from 12-01-2015 to 15-01-2015 in the ‘Entity’ block. The visualisation modes are ‘Bar’ chart and ‘Table’. The ‘Description’ data type brings more detailed information about the indicator results. From this indicator, we can know that Benoit has acquired a ‘good’ level (=1) compared with his original level (‘not very good’ = –1) in MySQL and he sets the target level of MySQL is ‘very good’ (=2). The level of documentation skill is achieved ‘neutral’ (=0) from the ‘not very good’ level (=–1) and his target level of documentation is ‘good’ (=1). So he needs to progress in MySQL and documentation.

Figure 5 Creation of a ‘knowledge development’ indicator (see online version for colours)

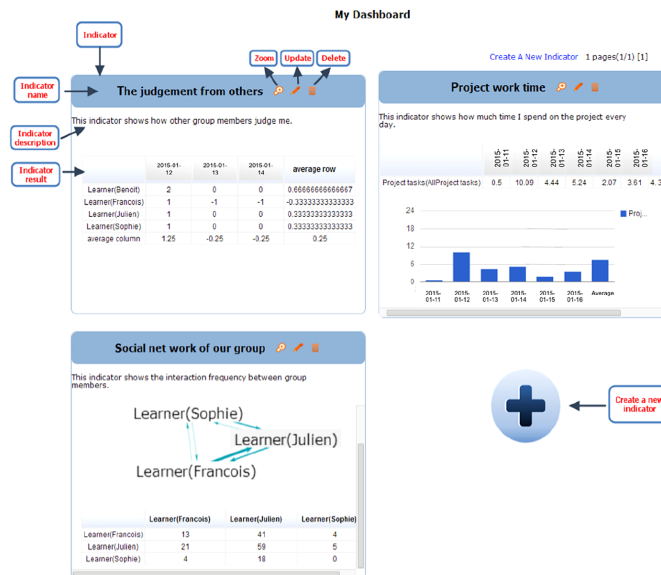


4.3 Customised dashboard

The objective of the dynamic dashboard is to help learners to create and manage customisable indicators. It allows learners to explore the activity and reporting traces and to choose the useful visualisation(s).

All the indicators that have been created by learners are presented in their own dashboards. For instance, in Figure 6, the dashboard is composed of three indicators. Each one is composed of three parts: the indicator name, the indicator description and the graphical part. The learners can manage the indicators by zooming, updating or deleting information. They can observe these indicators separately or on the same screen in order to compare them. This feature could be useful to discover problems in their projects or learning, such as spending little time in the project, working alone in the group, progressing too slowly in the project schedule and so on.

Figure 6 Dashboard interface (see online version for colours)



The three indicators presented in the dashboard in Figure 6 focused on different aspects. The first indicator 'The judgements from others' presents the results of the peer-evaluation gained from the other group members as well as the average evaluation. The range of the evaluation is from -2 (not at all good) to +2 (very good). Hence, we can find the average evaluation (=0.25) is neutral. With the second indicator 'Project work time', we can find the learner spent the most time (=10.09 h) on the project at 12th Jan 2015 and the average working time is 4.3 h from 11th Jan 2015 to 16th Jan 2015. The third indicator 'Social network of our group' presents the interaction frequencies between three group members: Sophie, Francois and Julien. We can observe that Julien contacts Francois frequently while Sophie seldom contacts her teammates.

The indicators can depict some important information from the traces, which would be difficult for learners to discover by themselves without a visual representation. Finally, from the dashboard screen, learners can create a new indicator by clicking on the button 'Create a new indicator'. They are redirected to the interface dedicated to the indicator creation or modification (Figure 3).

4.4 Discussion

The DDART tool has been designed to meet the users' need to create their own indicators and aggregate them into dashboards in order to have a global and personalised vision on an educational project. We have seen in previous sections, through examples, that this tool can actually be used to create rich indicators and manage them. Technical validation is done; the system provides relevant features to create a variety of indicators and customised dashboards. The tool has been designed to require no particular skill in computer science or data analysis, it is not necessarily easy for a student to master it. But usability issues have to be studied for this kind of system in order to test if our design choices are right. Indeed, many studies relate user problems linked with lack of information feedback and difficulty with indicator exploitation due to bad design of the interfaces and interaction. This test must be done before proposing DDART to students in real PBL contexts and measure its real utility and efficiency to support SRL purpose. We present in the next section an evaluation focusing on usability issues.

5 Usability evaluation

5.1 Participants, material, procedure of the experiment

We have conducted an experiment with 12 learners in order to test the usability of DDART. The experiment was semi-controlled and the participants had to follow a scenario to complete different tasks with DDART. By analysing the experiment traces and the feedbacks of the survey, we concluded with some advantages and disadvantages of DDART.

The participants (five women and seven men), ranging in age from 21 to 28 years old, voluntarily participated in this experiment. All the participants are used to work with others in PBL activities and they have already used collaborative and distance working tools and dashboards, including Moodle. They never used DDART before.

All of them were assigned to the same experimental scenario. We have simulated the traces left in the PBLMS by a group of learners as if they carried out a project for a week, for example: messages posted in the chat, documents written in the wiki, messages read in the forum, activities recorded in the reporting tool and indicators created in the dashboard. We explained to each participant that they took the role of Benoit and belonged to this PBL group.

The participants were required to use a computer installed with a browser to access to the internet. All the participants were supplied with an online scenario that guided the experiment. Indeed, during the experiment, they had to answer various questions about predefined indicators presented into the dashboard in order to test their readability. They then had to build two specific indicators in order to test the easiness of use and learnability of the indicator design interface. After the experiment, they had to answer an online survey composed of the 'SUS' (System Usability Scale) questionnaire (Brooke, 1996) and of several open questions to test the general usability of DDART. The participants' activity traces were also recorded.

We distinguished two groups of participants, according to the condition of the experiment:

- The distance group ($n = 7$): they did the experiment at distance, without any help during the scenario.
- The presence group ($n = 5$): they did the experiment in the presence of a person to help them immediately when they were in trouble with the system.

The feedbacks from the questionnaires and the participants' traces are analysed and discussed in the following parts.

5.2 General usability

We calculated the average scores of the SUS questionnaire for the two groups. The score for the remote group is 53.93 and for the group at the presence is 54.50. The average score is 54.17. We observed that the SUS scores of the two groups are almost the same. According to Bangor, Kortum and Miller (2008), the SUS score of DDART is in the range of 'marginal' (that means between 'ok' and 'good') and highlights some problems in the uses. The analyses of readability, ease of use and learnability give more details about the reasons of this score.

5.3 Readability

The readability is measured according to the number of right answers given by the participants on questions related to the group behaviour like "who is the most active in the group, how long is your weekly average work time". The group behaviour is represented into predefined indicators proposed in the dashboard at the beginning of the experiment. We calculated the average accuracy of the answers in Table 1.

Table 1 Average accuracy of the answers on the predefined indicators

<i>Distant group</i>	<i>Present group</i>
90.11%	90.77%

From the above table, the average accuracies of the two groups are almost the same, which are up to 90%. For the present group, the assistant has never been called for help. Namely, most of the participants were able extract the right information from the indicators and the indicators were understood by most of the participants.

5.4 Ease of use and learnability

In order to evaluate the ease of use and the learnability of the dashboard, we analysed the experimental traces related to the creation of two customisable indicators and obtained the results presented in Table 2. For the two indicators the users had to create (named respectively I1 and I2), we mainly focus on 4 indexes:

- 1 Success Ratio of creating an Indicator (SRI): represents the ratio C/N where C is the number of participants that can create the specified indicator successfully and N is the group size.
- 2 Average Time of creating an Indicator (ATI): is the mean time spent to create an indicator.
- 3 Efficiency I: is a ratio of the success rate to the average time (Albert and Tullis, 2013). Basically, it expresses the task success per unit of time. In our case, we use minute as the unit of time. The higher the value is, the more efficient the system is.
- 4 Average Invalid operations ratio of creating an Indicator (AII): is the mean of invalid operations to create an indicator. The valid operations are selecting the correct indicator parameters and deleting the wrong parameters.

Table 2 Ease of use and the learnability of the dashboard

		<i>Distant group</i>		<i>Present group</i>		
I1	SRI	57%		100%		
	ATI	12'31"		7'39"		
	E	4.56		13.07		
	AII	70.7%	AII 'X/Y entities'	27.9%	83.71%	AII 'X/Y entities'
			AII 'Data type'	35%	AII 'Data type'	13.3%
			AII 'Visualisations'	11.9%	AII 'Visualisations'	10%
I2	SRI	83%		100%		
	ATI	6'55"		5'04"		
	E	11.99		19.76		
	AII	74.27%	AII 'X/Y entities'	27.7%	89.28%	AII 'X/Y entities'
			AII 'Data type'	5.56%	AII 'Data type'	0%
			AII 'Visualisations'	0%	AII 'Visualisations'	0%

5.4.1 Ease of use

By observing the indexes of success ratio, average time and efficiency of the distance group presented in Table 2, we observe that these three indexes are not positive. Comparing the two groups, the presence group spent less time on the indicator creation

(7'39") than the other one (12'31"). The success ratio of indicator creation and the efficiency value of the presence group (respectively, 100% and 13.07) are higher than the distance group (respectively, 57% and 4.56). More precisely, we can observe that, in both groups, the invalid operations are highly related to the choice of the 'X/Y entities'. For I1, AII 'X/Y entities' is at 16 and 11.1% in the presence group and 27.9 and 27.7% in the distance group.

One explanation is that the specification of the filter is coupled with the choice of the 'X/Y entities'. Indeed, many assistance interventions for the presence group were due to trouble in the manipulation of the filtering function. This function is hidden in the interface (visible when the user click on a + button) and so not easily identified by the users when they had to build I1 and I2. Another explanation is that the participants did not carry out the project in ecological conditions. All the experiment data are simulated and were integrated in the system before the experiment. It is difficult for users to understand the significance of entities, how to manipulate them and the data linked to them. We can suppose that if the users had produced the data, it would be more intuitive for them to understand their significance. The last explanation is that the indicator design requires being able to choose the appropriate indicator, to formalise it and then to select the right representation. We believe that users did not have all these skills.

Moreover, with a technical assistance, results are really better and 100% of the users of the presence group succeeded in creating the indicators. So we can conclude that the principle of the indicators design is not intuitive, but novice participants succeed in creating indicators, if they have some technical assistance.

5.4.2 Learnability

The learnability assesses how much time or effort is required to achieve a maximum efficiency (Albert and Tullis, 2013). Considering the two groups, we observe that the four indexes of the second indicator are much better than the first indicator. The success ratio increases, the average time is shortened, the valid operation ratio is improved and the efficiency is also advanced.

Regarding the number of invalid operations done to create I1 and I2, we can observe a decrease for the operation linked with data value selection (from 35 to 5.6% for the distance group and 13.3 to 0% for the presence group) and visualisation (from 11.9 to 0% for distance group and 10 to 0% for the presence group). We can conclude that the visualisation interactions are easily learnable and that data value selection interactions are easy to understand with practice. But the operations required for the selection of the IT set, the selection of X/Y entities and the filters are not well understood for the design of I1 and I2, both for the distance group (with scores of 27.9 and 27.7%) and presence group (with 16 and 11.1%) in which users have technological assistance. The interactions proposed for these functions are not intuitive and not easy to understand.

So we can conclude that the interactions proposed for the entities choice and the filtering functions have to be redesigned. The other interactions like data value selection and visualisation seems easy to learn. We have to conduct other experimentations on a longer period to confirm this tendency.

6 Conclusion and future works

In this research, we mainly develop an architecture of PBLMS used to help learners to gain in self-regulation skills required in PBL. A PBLMS is composed of four modules: trace collection, trace integration, indicator calculation and indicator visualisation. In the trace collection module, we propose to help learners to reflect on their project and learning processes by recording their activities, assessments and judgements manually (self-report). The trace integration module is based on a uniform integrated trace model to integrate the activity traces and the reporting traces. It gives learners the possibility to explore the two types of traces as a uniform set and so to create personalised indicators. The two other modules enable learners to set different indicator parameters and visualisations. This whole process supports the three processes of SRL (forethought, performance and self-reflection) and the creation of indicators in all dimensions (cognitive, motivational and behavioural).

However, there are some limitations of the proposed PBLMS. First, a PBLMS depends on the associated LMS and cannot be integrated to other LMS easily. For example, a PBLMS needs to have access to some data from the LMS database. An issue to be considered is the adoption of a modular and generic approach for the design of the PBLMS so as to be able to integrate it to various LMS. Second, the semi-structured sentences are predefined and tutors or learners cannot add, modify or delete sentence models. However users of a PBLMS can conduct different projects, with different goals and contents. So the semi-structured sentences are not suitable to all projects. Another issue is to give the possibility to tutors and/or learners to manage the sentence models for their own project.

Based on the general architecture of PBLMS, we have developed a system named DDART. We observed that DDART is useful in the context of projects, because it supplies learners with functions to add information, thoughts and reflections on the project into the system and to monitor their projects by creating personalised indicators. The reporting tool supports learners' reflection and enables them to input information to create rich personalised indicators in their customisable dashboard. They can so monitor their projects by their own way and be engaged in the monitoring process. A comparative evaluation of the technical possibilities of DDART shows that many rich indicators used in TEL are reproducible. The usability evaluation shows that the general design is rather good except for the filtering function, and DDART is easy to use but not completely intuitive: a technical assistance has to be given to facilitate the first uses.

However, one limitation of the evaluation is that it did not target the usefulness of our approach to support SRL processes. Such an evaluation should be conducted on a long-term period, with specific measures to observe SRL processes and the acquisition of SRL skills. We plan to carry out a long-term experiment to test the utility of DDART in a real PBL course. We will more precisely analyse the way DDART is used with mixed (quantitative and qualitative) methods combining interaction traces, observation of the indicators built by students, surveys and interviews. Based on these data, we will deduce students' strategies (e.g. why they choose to focus and assess specific activities, how they judge them), the way they develop SRL competencies and their impact on the learning process.

Moreover, DDART was developed from the learners' points of views and we focused on how to support learner's SRL in PBL. However, tutors' self-regulation skills enhancement is ignored by most existing researches. We think that this issue should be

addressed, because tutors have to monitor the PBL processes and to help learners to reach their learning goals by themselves and consciously. We plan to develop a system for tutors to improve their self-regulation skills by considering the differences between learners' and tutors' self-regulation processes.

Reference

- Albert, W. and Tullis, T. (2013) *Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics*, Morgan Kaufmann, Waltham, USA.
- Bangor, A., Kortum, P. and Miller, J. (2008) 'An empirical evaluation of the system usability scale', *International Journal of Human-Computer Interaction*, Vol. 24, No. 6, pp.574–594.
- Brooke, J. (1996) 'SUS-A quick and dirty usability scale', *Usability Evaluation in Industry*, Vol. 189, No. 194, pp.4–7.
- Buck Institute for Education. (2014) *What is Project Based Learning?* [online], Buck Institute for Education. Available at: http://bie.org/about/what_pbl (access 9 December 2015).
- Chang, M-M. (2007) 'Enhancing web-based language learning through self-monitoring', *Journal of Computer Assisted Learning*, Vol. 23, pp.187–196.
- Charleer, S., Santos, J.L., Klerkx, J. and Duval, E. (2014) 'LARAe: learning analytics reflection & awareness environment', *ARTEL14: Proceedings of the 4th Workshop on Awareness and Reflection in Technology-Enhanced Learning*, Graz, Austria, pp.85–87, [online] <http://ceur-ws.org/Vol-1238/>.
- Després, C. and Leroux, P. (2002) 'Un outil de suivi synchrone pour la formation à distance', *TICE 2002: Technologies de l'Information et de la Communication dans les Enseignements d'ingénieurs et dans l'industrie*, Lyon, France, pp.315–320, [online] <http://docinsa.insa-lyon.fr/tice/2002/ca/ca051.html>.
- English, M.C. and Kitsantas, A. (2013) 'Supporting student self-regulated learning in problem- and project-based learning', *Interdisciplinary Journal of Problem-Based Learning*, Vol. 7, No. 2, pp. 128–150 [online] <http://dx.doi.org/10.7771/1541-5015.1339> (access 9 December 2015).
- Ferguson, R., Buckingham Shum, S. and Deakin Crick, R. (2011) 'EnquiryBlogger: using widgets to support awareness and reflection in a PLE setting', *1st Workshop on Awareness and Reflection in Personal Learning Environments*, Southampton, United Kingdom, pp.28–33.
- Fessl, A., Wesiak, G. and Luzhnic, G. (2014) 'Application overlapping user profiles to foster reflective learning at work', *ARTEL14: Proceedings of the 4th Workshop on Awareness and Reflection in Technology-Enhanced Learning*, pp.51–64, [online] <http://ceur-ws.org/Vol-1238/> (access 9 December 2015).
- Gendron, É. (2010) *Cadre Conceptuel pour l'élaboration d'indicateurs de collaboration à partir des traces d'activité*, Unpublished PhD Thesis, Université Claude Bernard-Lyon I, France.
- Glaser-Zikuda, M., Fendler, J., Noack, J. and Ziegelbauer, S. (2011) 'Fostering self-regulated learning with portfolios in schools and higher education', *Orbis Scholae*, Vol. 5, No. 2, pp.67–78.
- Govaerts, S., Verbert, K., Duval, E. and Pardo, A. (2012) 'The student activity meter for awareness and self-reflection', *CHI'12: Extended Abstracts on Human Factors in Computing Systems*, New York, USA, pp.869–884.
- Graf, S., Ives, C., Rahman, N. and Ferri, A. (2011) 'AAT: a tool for accessing and analysing students' behaviour data in learning systems' *1st International Conference on Learning Analytics and Knowledge (LAK 11)*, Banff, AB, Canada, pp.174–179.
- Gueraud, V. and Cagnat, J.M. (2004) 'Suivi à distance de classe virtuelle active', *TICE 2004: Technologies de l'Information et de la Connaissance dans l'Enseignement Supérieur et l'Industrie*, Compiègne, France, pp.377–383.
- Janssen, J., Erkens, G. and Kirschner, P.A. (2011) 'Group awareness tools: it's what you do with it that matters', *Computers in Human Behavior*, Vol. 27, No. 3, pp.1046–1058.

- Järvelä, S. and Hadwin, A.F. (2013) 'New frontiers: regulating learning in CSCL', *Educational Psychologist*, Vol. 48, No. 1, pp.25–39.
- Ji, M., Michel, C., Lavoué, E. and George, S. (2013) 'An architecture to combine activity traces and reporting traces to support self-regulation processes', *ICALT 2013: 13th IEEE International Conference on Advanced Learning Technologies*, Beijing, China, pp.87–91.
- Kimmerle, J. and Cress, U. (2008) 'Group awareness and self-presentation in computer supported information exchange', *International Journal of Computer-Supported Collaborative Learning*, Vol. 3, No. 1, pp.85–97.
- Lajoie, S.P. and Lu, J. (2011) 'Supporting collaboration with technology: does shared cognition lead to co-regulation in medicine?', *Metacognition and Learning*, Vol. 7, No. 1, pp.45–62.
- Lamprey, W. and Fayek, A. (2012) 'Developing a project status dashboard for construction project progress reporting', *International Journal of Architecture, Engineering and Construction*, Vol. 1, No. 2, pp.112–120.
- Littlejohn, A., Margaryan, A. and Milligan, C. (2009) 'Charting collective knowledge: supporting self-regulated learning in the workplace', *ICALT 2009: Ninth IEEE International Conference on Advanced Learning Technologies, 2009*, Riga, Latvia, pp.208–212.
- Loboda, T.D., Guerra, J., Hosseini, R. and Brusilovsky, P. (2014) 'Mastery grids: an open source social educational progress visualization', *EC-TEL 2014: Proceedings of the Ninth European Conference on Technology Enhanced Learning*, Toledo, Spain, pp.235–248.
- May, M., George, S. and Prévôt, P. (2011) 'TrAVis to enhance students' self-monitoring in online learning supported by computer-mediated communication tools', *Computer Information Systems and Industrial Management Applications*, Vol. 3, pp.623–634.
- Mazza, R. and Dimitrova, V. (2007) 'CourseVis: a graphical student monitoring tool for supporting instructors in web-based distance courses', *International Journal of Human-Computer Studies*, Vol. 65, No. 2, pp.125–139.
- Michel, C. and Lavoué, E. (2011) 'KM and Web 2.0 methods for project-based learning', in Ifenthaler, D. et al. (Ed.): *Multiple Perspectives on Problem Solving and Learning in the Digital Age*, Springer-Verlag, New York, USA, pp.49–63.
- Michel, C., Lavoué, E. and Pietrac, L. (2012) 'A dashboard to regulate project-based learning' *EC-TEL 2012: Seventh European Conference on Technology Enhanced Learning*, Saarbrücken, Germany, pp.250–263.
- Narciss, S., Proske, A. and Koerndle, H. (2007) 'Promoting self-regulated learning in web-based learning environments', *Computers in Human Behavior*, Vol. 23, No. 3, pp.1126–1144.
- Pintrich, P.R. (2004) 'A conceptual framework for assessing motivation and self-regulated learning in college students', *Educational Psychology Review*, Vol. 16, No. 4, pp.385–407.
- Reffay, C. and Chanier, T. (2003) 'How social network analysis can help to measure cohesion in collaborative distance-learning', *CSCL 2003: Proceedings of the International Conference on Computer Support for Collaborative Learning 2003*, Bergen, Norway, pp.343–352.
- Sangin, M., Molinari, G., Nassli, M-A. and Dillenbourg, P. (2011) 'Facilitating peer knowledge modeling: effects of a knowledge awareness tool on collaborative learning outcomes and processes', *Computers in Human Behavior*, Vol. 27, No. 3, pp.1059–1067.
- Winne, P.H., Hadwin, A.F. and Gress, C. (2010) 'The learning kit project: software tools for supporting and researching regulation of collaborative learning', *Computers in Human Behavior*, Vol. 26, No. 5, pp.787–793.
- Zimmerman, B. (1989) 'Models of self-regulated learning and academic achievement', in Zimmerman, B. et al. (Ed.): *Self-Regulated Learning and Academic Achievement*, Springer, New York, USA, pp.1–25.
- Zimmerman, B. (2002) 'Becoming a self-regulated learner: an overview', *Theory into Practice*, Vol. 41, No. 2, pp.64–70.