



HAL
open science

Exploiting Activity Traces and Learners' Reports to Support Self-Regulation in Project-based Learning

Min Ji

► **To cite this version:**

Min Ji. Exploiting Activity Traces and Learners' Reports to Support Self-Regulation in Project-based Learning. Computers and Society [cs.CY]. INSA de Lyon, 2015. English. NNT: . tel-01212256v1

HAL Id: tel-01212256

<https://hal.science/tel-01212256v1>

Submitted on 6 Oct 2015 (v1), last revised 6 Oct 2015 (v2)

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.

Dissertation

Exploiting Activity Traces and Learners' Reports to Support Self-Regulation in Project-based Learning

Submitted to
National Institute of Applied Sciences of Lyon
(Institut National des Sciences Appliquées de Lyon)

For
The doctoral degree in Computer Science

Doctoral school:
Informatics and Mathematics

By
Min JI

Defense on the 27th of April 2015 in front of the commission board

Jury

Reporters:	Eric Bruillard	Professor (ENS de Cachan)
	Ang élique Dimitracopoulou	Professor (University of the Aegean)
Examiners:	Youssef Amghar	Professor (INSA de Lyon)
	Christophe Kolski	Professor (Université de Valenciennes et du Hainaut Cambr ésis)
	Marie-H é ène Abel	Professor (Université de Technologie de Compi ègne)
Supervisors:	Sebastien George	Professor (Université du Maine)
	Christine Michel	Associate professor (INSA de Lyon)
	Elise Lavou é	Associate professor (Université Jean Moulin Lyon 3)

INSA Direction de la Recherche - Ecoles Doctorales - Quinquennal 2011-2015

SIGLE	ECOLE DOCTORALE	NOM ET COORDONNEES DU RESPONSABLE
CHIMIE	CHIMIE DE LYON http://www.edchimie-lyon.fr Sec : Renée EL MELHEM Bat Blaise Pascal 3 ^e etage 04 72 43 80 46 Insa : R. GOURDON secretariat@edchimie-lyon.fr	M. Jean Marc LANCELIN Université de Lyon – Collège Doctoral Bât ESCPE 43 bd du 11 novembre 1918 69622 VILLEURBANNE Cedex Tél : 04.72.43 13 95 directeur@edchimie-lyon.fr
E.E.A.	ELECTRONIQUE, ELECTROTECHNIQUE, AUTOMATIQUE http://edeea.ec-lyon.fr Sec : M.C. HAVGOUDOUKIAN eea@ec-lyon.fr	M. Gérard SCORLETTI Ecole Centrale de Lyon 36 avenue Guy de Collongue 69134 ECULLY Tél : 04.72.18 60.97 Fax : 04 78 43 37 17 Gerard.scorletti@ec-lyon.fr
E2M2	EVOLUTION, ECOSYSTEME, MICROBIOLOGIE, MODELISATION http://e2m2.universite-lyon.fr Sec : Safia AIT CHALAL Bat Atrium- UCB Lyon 1 04.72.44.83.62 Insa : S. REVERCHON Safia.ait-chalal@univ-lyon1.fr	Mme Gudrun BORNETTE CNRS UMR 5023 LEHNA Université Claude Bernard Lyon 1 Bât Forel 43 bd du 11 novembre 1918 69622 VILLEURBANNE Cedex Tél : 06.07.53.89.13 e2m2@univ-lyon1.fr
EDISS	INTERDISCIPLINAIRE SCIENCES-SANTE http://www.ediss-lyon.fr Sec : Safia AIT CHALAL Bat Atrium – UCB Lyon 1 04 72 44 83 62 Insa : Safia.ait-chalal@univ-lyon1.fr	Mme Emmanuelle CANET-SOULAS INSERM U1060, CarMeN lab, Univ. Lyon 1 Bâtiment IMBL 11 avenue Jean Capelle INSA de Lyon 696621 Villeurbanne Tél : 04.72.68.49.09 Fax :04 72 68 49 16 Emmanuelle.canet@univ-lyon1.fr
INFOMATHS	INFORMATIQUE ET MATHEMATIQUES http://infomaths.univ-lyon1.fr Sec :Renée EL MELHEM Bat Blaise Pascal 3 ^e etage infomaths@univ-lyon1.fr	Mme Sylvie CALABRETTO LIRIS – INSA de Lyon Bat Blaise Pascal 7 avenue Jean Capelle 69622 VILLEURBANNE Cedex Tél : 04.72. 43. 80. 46 Fax 04 72 43 16 87 Sylvie.calabretto@insa-lyon.fr
Matériaux	MATERIAUX DE LYON http://ed34.universite-lyon.fr Sec : M. LABOUNE PM : 71.70 –Fax : 87.12 Bat. Saint Exupéry Ed.materiaux@insa-lyon.fr	M. Jean-Yves BUFFIERE INSA de Lyon MATEIS Bâtiment Saint Exupéry 7 avenue Jean Capelle 69621 VILLEURBANNE Cedex Tél : 04.72.43 71.70 Fax 04 72 43 85 28 Ed.materiaux@insa-lyon.fr
MEGA	MECANIQUE, ENERGETIQUE, GENIE CIVIL, ACOUSTIQUE http://edmega.universite-lyon.fr/ Sec : M. LABOUNE PM : 71.70 –Fax : 87.12 Bat. Saint Exupéry mega@insa-lyon.fr	M. Philippe BOISSE INSA de Lyon Laboratoire LAMCOS Bâtiment Jacquard 25 bis avenue Jean Capelle 69621 VILLEURBANNE Cedex Tél : 04.72 .43.71.70 Fax : 04 72 43 72 37 Philippe.boisse@insa-lyon.fr
ScSo	ScSo* http://recherche.univ-lyon2.fr/scso/ Sec : Viviane POLSINELLI Brigitte DUBOIS Insa : J.Y. TOUSSAINT viviane.polsinelli@univ-lyon2.fr	Mme Isabelle VON BUELTZINGLOEWEN Université Lyon 2 86 rue Pasteur 69365 LYON Cedex 07 Tél : 04.78.77.23.86 Fax : 04.37.28.04.48 isavonb@gmail.com

*ScSo : Histoire, Géographie, Aménagement, Urbanisme, Archéologie, Science politique, Sociologie, Anthropologie

Exploiting Activity Traces and Learners' Reports to Support Self-Regulation in Project-based Learning

Abstract

Project-based Learning (PBL) is a learner-oriented instructional method, which enables learners to carry out challenging and authentic projects by thorough investigations. PBL affords learners the opportunities to organize and plan the project, to collaborate with peers and to look for the resources and guidance to achieve the project goals. However, PBL is difficult to implement successfully because learners often lack of the self-regulation skills required to monitor, reflect, manage and assess their project activities and learning. Self-Regulated Learning (SRL) can train learners to gain these skills. However, most learning systems used in PBL focus on providing rich learning materials to the learners but rarely offer possibilities to monitor and analyze their project and learning processes. The main goal of this thesis is to support SRL during PBL situations.

We propose a general architecture of Project-based Learning Management System (PBLMS), which help learners to understand how to regulate their learning activities during the projects. This general architecture integrates an existing Learning Management System (LMS) and two tools we propose: a reporting tool and a dynamic dashboard. The reporting tool enhances learners' reflective processes by leading them to describe their non-instrumented activities, their reflections and assessments on the project activities based on semi-structured sentences. The system can record automatically the activity traces of the users' interactions with the LMS, the reporting tool and the dashboard. These activity traces are merged with the self-reporting data so that indicators can be calculated basing on this entire information. The dynamic dashboard supports learners in creating customizable indicators. Learners can specify the data to take into account, the calculation and the visualization modes. We implemented this theoretical proposition with the development of the DDART (Dynamic Dashboard based on Activity and self-Reporting Traces) platform that integrates the reporting tool and the dynamic dashboard.

To evaluate the proposition, we firstly test the ability of DDART to recreate a large sample of indicators that are proposed in existing researches about the analysis of activities, cognition, emotion and social network. Furthermore, an experiment was conducted to evaluate the usability and utility of DDART. According to the results of this experiment, we found that DDART supports learners' reflections on the way they carry out the project and provides them with the opportunities to monitor their activities and learning, even if the indicator creation could be difficult for the novices.

Key words: Project-based Learning, Self-Regulated Learning, Project-based Learning Management System, activity trace, self-reporting, dynamic dashboard, indicator

Table of contents

Abstract.....	i
Table of contents.....	ii
List of figures.....	iv
List of tables	vi
1 Introduction	1
1.1 Research context.....	1
1.2 Research questions	3
1.3 Thesis organization.....	3
2 State of the art	5
2.1 Definitions and characteristics of Project-based Learning and Self-Regulation Learning.....	5
2.1.1 Definition of PBL.....	5
2.1.2 Definition of SRL and SRL tools	7
2.2 E-learning systems used to support SRL in PBL.....	11
2.3 Traces and tools used to support SRL in PBL	16
2.3.1 Activity traces in learning environments	16
2.3.2 Reporting traces in learning environments.....	17
2.3.3 Hybrid traces in learning environments	19
2.3.4 Discussion	21
2.4 Trace presentation: simple dashboard and dynamic dashboard	22
2.4.1 Simple dashboards with predefined indicators.....	23
2.4.2 Dynamic dashboards with customizable indicators	31
2.4.3 Discussion	33
2.5 Conclusion.....	35
3 A Project-based Learning Management System.....	37
3.1 Proposition of a general framework of PBLMS.....	37
3.2 Indicator management	40
3.3 Trace collection	42
3.3.1 Activity traces collection.....	43
3.3.2 Reporting traces collection.....	45
3.4 Trace integration.....	48
3.5 Indicator calculation	50
3.5.1 Data selection	51
3.5.2 Data types	53
3.5.3 Calculation operators	58
3.6 Indicator visualization	60
3.7 Conclusion.....	61
4 Implementation of PBLMS	63

Table of contents

4.1	General architecture of PBLMS	63
4.2	Reporting tool.....	67
4.3	Dynamic dashboard.....	71
4.4	Conclusion.....	79
5	Illustration of use, experiment and result analysis	81
5.1	Test of the ability of DDART to create existing indicators	81
5.1.1	Activity indicators.....	81
5.1.2	Social indicators.....	90
5.1.3	Cognitive indicators	93
5.1.4	Affective indicators.....	95
5.1.5	Discussion	97
5.2	Experiment	98
5.2.1	Experiment research questions.....	98
5.2.2	Participants.....	99
5.2.3	Materials.....	99
5.2.4	Procedure.....	100
5.2.5	Results analysis	101
5.3	Conclusion.....	114
6	Conclusion and Perspectives.....	115
6.1	Conclusion.....	115
6.2	Perspectives	116
Annex	119
A.	The data models of the activity traces in Moodle tools.....	119
B.	The data models of the activity traces in reporting tool and dashboard.....	122
C.	The DTD and XML used in reporting tool.....	123
D.	Examples of visualization modes.....	128
E.	The experiment scenario	133
F.	The experiment survey.....	136
Bibliography	139

List of figures

Figure 2-1 The dashboard of Study desk (Narciss <i>et al.</i> 2007).....	24
Figure 2-2 The dashboard of Feeler (Durall and Toikkanen 2013).....	24
Figure 2-3 The dashboard of NAVI (Santos <i>et al.</i> 2013)	25
Figure 2-4 The dashboard of Radar and Reflector (Phielix <i>et al.</i> 2010).....	26
Figure 2-5 The dashboard of LARAE (Charleer <i>et al.</i> 2014).....	27
Figure 2-6 The dashboard of the Mastery Grids (Loboda <i>et al.</i> 2014)	28
Figure 2-7 The dashboard of Moodog (Zhang <i>et al.</i> 2007)	29
Figure 2-8 The dashboard of GISMO (Mazza and Milani 2004)	29
Figure 2-9 The dashboard of Student Activity Meter (Govaerts <i>et al.</i> 2012).....	30
Figure 2-10 The dashboard of Navi Surface (Santos <i>et al.</i> 2013)	31
Figure 2-11 The dashboard of AAT (Graf <i>et al.</i> 2011).....	32
Figure 2-12 The dashboard of GINDIC (Gendron 2010)	32
Figure 2-13 The dashboard of TBS-IM (Djouad and Mille 2010)	33
Figure 3-1 The components of PBLMS	37
Figure 3-2 The framework of PBLMS	39
Figure 3-3 The process of creating a new indicator	41
Figure 3-4 The process of updating an indicator.....	41
Figure 3-5 The architecture of activity trace collection process	43
Figure 3-6 An example of the activity “sending a chat message”	44
Figure 3-7 The entity-relationship schema of “sending a chat message”	44
Figure 3-8 The architecture of the reporting tool.....	46
Figure 3-9 The class model of the reporting traces.....	47
Figure 3-10 The process of trace integration.....	49
Figure 3-11 The crossing process	52
Figure 3-12 The filtering process	53
Figure 3-13 The frequency process	54
Figure 3-14 The time interval process	55
Figure 3-15 The content process	56
Figure 3-16 The time span process	56
Figure 3-17 The description process	57
Figure 3-18 The simple operator process.....	59
Figure 3-19 The complex calculation operator process.....	60
Figure 3-20 The process of indicator visualization.....	61
Figure 4-1. The main interface of PBLMS	64
Figure 4-2 The main interfaces of the Moodle tools.....	64
Figure 4-3 The architecture of DDART.....	65
Figure 4-4 The automatic traceable activities in PBLMS	66

List of figures

Figure 4-5 The interface of writing a new report.....	68
Figure 4-6 The interfaces of reporting tool.....	68
Figure 4-7 The report list written by a specific group member	69
Figure 4-8 Modification of the report contents.....	70
Figure 4-9 To reply the comments related to a report written by the learner	70
Figure 4-10 The report list written by other group members.....	70
Figure 4-11 The content and comments of a report written by another group member	71
Figure 4-12 The dashboard interface	72
Figure 4-13 The interface to create an indicator	73
Figure 4-14 The parameter blocks in the interface to create an indicator	74
Figure 4-15 The example of creating indicator.....	76
Figure 4-16 the interface to save an indicator.....	78
Figure 5-1 The visualization of “students login overview” indicator (Mazza and Milani 2004).....	82
Figure 5-2 The interface to create the “students login overview” indicator (1) ..	83
Figure 5-3 The interface to create the “students login overview” indicator (2) ..	84
Figure 5-4 The visualization of “users classification” indicator (Bratitsis and Dimitracopoulou 2006).....	85
Figure 5-5 The interface to create the “user classification” indicator	86
Figure 5-6 The visualization of “time spent on activities” indicator (Santos <i>et al.</i> 2012)	87
Figure 5-7 The interface to create the “time spent on activities” indicator	88
Figure 5-8 The visualization of “time spent on tools” indicator.....	89
Figure 5-9 The interface to create the “knowself rescourseblock” indicator.....	89
Figure 5-10 The visualization of the “social network of forum messages” indicator (Reffay and Chanier 2003)	90
Figure 5-11 The interface to create the “forum graph” indicator	91
Figure 5-12 The visualization of “peer assessment” indicator (Kennedy 2005) ..	92
Figure 5-13 The interface to create the “peer assessment” indicator.....	93
Figure 5-14 The visualization of “knowledge development” indicator (Michel <i>et</i> <i>al.</i> 2012).....	94
Figure 5-15 The interface to create the “knowledge development” indicator	95
Figure 5-16 The visualization of “mood view” indicator (Ferguson <i>et al.</i> 2011).....	96
Figure 5-17 The interface to create the “mood view” indicator	97
Figure 5-18 A comparison of SUS scores (Bangor <i>et al.</i> 2008).....	102
Figure 5-19 The opinions about learnability of the engineering student group ..	109
Figure 5-20 The opinions about learnability of the research student group	110
Figure 5-21 Which indicator is useful?.....	112

List of tables

Table 2-1 Pintrich’s SRL model (Pintrich 2000).....	8
Table 2-2 The micro-analytic assessment questions (Cleary and Zimmerman 2004)	9
Table 2-3 The architecture proposed by Laffey (Laffey <i>et al.</i> 1998)	14
Table 2-4 Comparison of the dashboards	33
Table 3-1 An example of the activity trace “sending a chat message”	45
Table 3-2 The class description	47
Table 3-3 Examples of integrated traces	50
Table 3-4 An example of crossing operation.....	51
Table 3-5 An example of filter operation	52
Table 3-6 An example of the frequency data type	53
Table 3-7 An example of the time interval data type.....	54
Table 3-8 An example of the content data type	55
Table 3-9 An example of the time span data type.....	56
Table 3-10 An example of the description data type	57
Table 3-11 An example of simple calculation operator.....	58
Table 3-12 An example of complex calculation operator.....	59
Table 4-1 The semi-structured sentence models.....	68
Table 4-2 Data dictionary of an indicator	78
Table 5-1 The scenario of the experiment	100
Table 5-2 The attributes of experimental traces	101
Table 5-3 The average scores of SUS	102
Table 5-4 The ease of use of the reporting tool	103
Table 5-5 The ease of use of the dashboard.....	106
Table 5-6 The average accuracy of the answers in survey	111

1 Introduction

1.1 Research context

The Buck Institute for Education, which focuses on the professional development and materials to support Project-based Learning (PBL), defined PBL 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 setting, projects are carried out under the complex situations and are derived by a science-based or authentic question or problem. Blumenfeld and Soloway (1991) described a precise processes of PBL like “*students pursue solutions to nontrivial problems by asking and refining questions, debating ideas, making predictions, designing plans and/or experiments, collecting and analyzing data, drawing conclusions, communicating their ideas and findings to others, asking new questions, and creating artifacts*”. This learner-centered strategy allows learners to collaborate autonomously during a period of time and produces the realistic artifacts or presentations (Thomas 2000), which represent a great understanding of a concept and deep learning. The tutors are important in the PBL. They are the conductors (direct learners to perform each step of PBL), questioners (guide learners to go further by asking questions), facilitators (create a positive collaborative environment in the group) and diagnosticians (intervene if the directions of the learners seem to be going badly off-course) (Smidts 2003). PBL is a key strategy for creating independent thinkers and learners. In a recent study (English and Kitsantas 2013), PBL was proved to facilitate knowledge acquisition and retention, supply an opportunity for learners to improve their skills in problem-solving, investigative activities, decision making, analyzing and evaluating information, thinking critically, working cooperatively and communicating effectively. By learning in PBL situations, learners become better researchers, problem solvers and higher-level thinkers (Gültekin 2005).

Despite these well-recognized advantages, some problems in PBL have been regarded as the main causes of the PBL failures, such as learners' low motivations, inappropriate behaviors, poor autonomic skills and the negative attitudes towards PBL (English and Kitsantas 2013). Therefore, it is important to support self-regulation in PBL, which can help to solve the above problems and increase the success ratio of PBL. Self-regulation skills support learners to take responsibility of their learning and aid them to improve their learning behaviors gradually. Self-Regulated Learning (SRL) was defined by Zimmerman (1989) as “*the degree to which learners are metacognitively, motivationally and behaviorally active participants in their own learning*”. Self-regulated learners enable to set their learning goals, plan, conduct, and then regulate and evaluate the learning processes independently (Narciss, Proske, and Koerndle 2007).

The development of internet technology provides the possibility to build the E-learning systems for learners and tutors, which can provide rich and functional learning environments for PBL. Sun *et al.* (2008) defined the E-learning system as

“a web based system that makes information or knowledge available to users or learners and disregards time restrictions or geographic proximity”. In E-learning systems, students can access not only to learning content materials, but also to plenty of tools to complete their projects, for example chats, forums, quizzes, video-conferences, blogs etc. Bouhnik and Marcus (2006) identified four advantages of E-learning systems: *“flexibility of the material and the time; accessibility to the material; visibility of the multimedia; availability of the data”*. However, most of the E-learning systems pay more attentions to supply rich learning materials to learners but rarely offer possibilities to monitor and analyze their learning processes and their project performances (Ji *et al.* 2014). Chen (2009) also pointed out one key problem: *“learners have to frequently interact with web-based learning systems even though they lack instructors to monitor their learning attitudes and behavior during learning processes”*. Hence, according to the learners' needs, it is important to provide the learners and the tutors with the means to respectively self-regulate and monitor the learning processes and projects.

Some methods have been proposed to train self-regulation skills, such as self-reporting questionnaires, structured interviews, teacher judgments, think-aloud protocols, error detection in tasks, trace methodologies and observation measures (Winne and Perry 2000). Trace methodologies are effective methods in PBL because traces are situated in the activities and provide information on learners' engagements in the activities (Perry and Winne 2006). Trace is defined as *“a collection of temporally situated observed elements”* (Djouad and Mille 2010). A trace can be a set of information explicitly linked to the learning activities (read the learning materials, create of a document, answer to a question), or a set of indices reflecting the learner's behaviors or characteristics (Settouti and Prie 2009). By manipulating the traces produced in PBL, the users (for example, learners, tutors, researchers) can extract the information about the project progresses and the learners' performances. Most of E-learning systems can capture the interactions between learners and the system, generate the traces automatically, and then use them to create indicators. But some projects require learners to do some activities out of the E-learning systems, for example, programming with NetBeans, writing reports in MS Word, collecting leaves in a forest, visiting factories etc. Unfortunately, the traces of these activities cannot be collected automatically by E-learning systems.

In order to enable users to understand the large amount of information contained in traces, most of the researchers decide to present them into the form of indicators. By observing the indicators, users can identify and outline the learners' behaviors. The indicators can be displayed in a dashboard. A dashboard is a visual interface used as a medium for reporting project progresses at a glance through indicators that are relevant to the success of the projects (Lamprey and Fayek 2012). Rasmussen, Bansal, and Chen (2009) has identified that the dashboards can not only improve the decision making and the performances, but also enhance the motivations of the group members. But, most of the dynamic dashboards (which allow users to create customizable indicators) are offered to tutors or researchers while the simple dashboards (which display the predefined indicators) are used for learners (Ji *et al.* 2014). Namely, learners accept passively the information supplied in the simple dashboards.

Hence, in our research, we want to propose a system to support self-regulation in PBL by allowing learners to plan their projects, reflect and record their activities accomplished in and out of the E-learning systems, and create customizable indicators to monitor their learning and their projects by the way they want.

1.2 Research questions

Based on the research context, we address the research questions as follow:

1 *What kind of system framework can support self-regulation in PBL?*

Our general research objective is to develop a system to support learners to use self-regulation skills during PBL. We hypothesize that it is useful to enable learners to plan their projects, reflect upon their project activities and monitor their behaviors by the way they want for the PBL objectives.

2 *What kinds of traces are useful to support self-regulation in PBL? How to collect them? Which data model can integrate the different types of traces?*

At first, it is necessary to identify the different types of traces and their characteristics because the traces are the foundation of the indicator creation. Then, we need to determine the different methods to collect them according to their different natures. At last, a common data model is necessary to integrate all different types of traces, which enables learners to explore all of the traces together.

3 *How to help learners to build a dynamic dashboard?*

We hypothesize that learners enable to self-regulate their learning and their projects if they use dynamic dashboard where the indicators are designed by themselves. The dashboard has to propose functionalities to offer learners the possibility (even if they have no computer background) to select the traces they are interested in, make calculation and visualize the results from multiple visualization modes.

1.3 Thesis organization

This thesis is composed of five chapters:

Chapter 2 is the state of the art. We review the existing literatures related to our research. At first, we give the definitions of PBL and SRL. We identify the main hindering factors of PBL implementation and some methods to improve self-regulation. Then, we study several E-learning systems that are designed for PBL and discuss their supporting levels of SRL. Next, we discuss how they use the traces. We distinguish two types of traces, automatically recorded and manually reported, and analyze their features and limitations. At last, we compare some simple dashboards with some dynamic dashboards based on several criteria (trace type, target user, customization, IT requirements) and draw the conclusion.

Based on the reviews of the existing researches in chapter 2, we propose a general framework in chapter 3, which can help learners to collect the different types

of traces and manipulate them to create customizable indicators. The framework is divided into four modules: trace collection, trace integration, indicator calculation and indicator visualization.

In chapter 4, we develop an implementation of the previous framework: DDART (Dynamic Dashboard based on Activity and Reporting Traces). We describe the technical architecture, the functions with the system interfaces and give examples.

In chapter 5, at first, we test the potential of DDART by observing how it can reproduce the indicators proposed in the literatures. Then, we carry out an experiment to test the usability and utility of the system. We discuss the advantages and the limitations of our system and propose some improvements.

In chapter 6, we draw some conclusions of our research and propose some perspectives.

2 State of the art

2.1 Definitions and characteristics of Project-based Learning and Self-Regulation Learning

At the beginning of this part, we give the definitions of Project-based Learning (PBL) and summary the main characteristics. PBL is a learner-centered method to help learners to generate new knowledge and improve their collaborative skills. However, there are some hindering factors of PBL implementation. In the second part, we study the definitions, models and the characteristics of SRL. Then, we introduce several methods that can support self-regulation.

2.1.1 Definition of PBL

Project-based Learning is “*a model that organizes learning around projects*” (Thomas 2000). Different PBL definitions are offered in different researches. Helm and Katz (2011) defined it as “*an instructional method that allows in-depth investigation of a topic instead of using a rigid lesson plan that directs a student down a specific path of learning outcomes or objectives*”. According to Jeremic (2009), “*PBL is a teaching and learning model that organizes learning around projects. Projects comprise complex tasks and activities that involve students in a constructive investigation that results in knowledge building.*” Donnelly and Fitzmaurice (2005) defined PBL as “*an individual or group activity that goes on over a period of time, resulting in a product, a presentation or a performance*”. Projects in PBL are different from conventional activities. Thomas (2000) defined five criteria for the project: “*PBL projects are central, not peripheral to the curriculum, PBL projects are focused on questions or problems that drive students to encounter (and struggle with) the central concepts and principles of a discipline, PBL projects involve students in a constructive investigation, PBL projects are student-driven to some significant degree, PBL projects are realistic, not school-like.*” From these definitions, we can summarize some features of PBL as follows:

1. ***Learners drive the project and learning.*** They need to take the responsibilities of the project and accomplish the project by planning, deciding their strategies, carrying out the project, assessing their performances, etc.
2. ***It is not “teacher telling”, but “learning by doing”.*** PBL projects are not teacher-led, scripted, or packaged (Thomas 2000). It is an investigation and inquiry process. This process results in the application and generation of new knowledge.
3. ***It ends with a presentation or product and the artifact is assessed.*** The artifact represents the knowledge that learners gain in the PBL. Tutors can

evaluate the effectiveness of PBL and learners can reflect on the learning progresses by assessing of the PBL achievements.

4. **The project is collaborative.** Learners need to work together in a group. In the project, learners can acquire mutual benefits from the brainstorm, discussion, collaboration with others.

In order to complete the authentic projects, students need to plan their learning, organize their researches, implement a multitude of learning strategies and apply different tools. Typically, complex projects increase learners' opportunities to think metacognitively and behave strategically because these tasks encourage learners to control challenges, evaluate multiple processes and products, and collaborate with peers (Perry and Winne 2006). Mergendoller *et al.* (2013) proposed a four-phases learning process of PBL:

1. **Project planning.** In this phase, learners need to determine the project scope and clarify the project issues by setting the project and learning goals. It helps them to know the gaps between the knowledge they have gained and the knowledge they need to get through the project. They organize all the resources (the knowledge they have gained, the human resources and the facilities: whiteboard, computer, chart, telephone and so on) and decide the strategies they want to apply in the project.
2. **Project launch.** Tutors (or learners themselves) begin to stimulate learners' interests and motivations that need to last for the latter 2 phases. One of the main reasons of the project failure is lack of motivations of learners. The project group needs to specify the project schedule, procedure, submission and work division.
3. **Guided inquiry and product creation.** Learners need to structure the project, decompose it into several small tasks, and implement adaptable resources to complete the project. During this phase, learners can improve their skills in communication, collaboration, documentation and so on. Tutors should pay attentions to the project development and learners' behaviors so as to intervene when some problems emerge.
4. **Project conclusion.** Learners present their project achievements. Tutors (or learners themselves) assess the project fruits and learners' performances during the whole project. Learners reflect on the learning processes, summarize their gains and losses that help them to do better in the future learning.

This self-driven, motivating approach helps learners to get valuable skills which can establish a solid foundation for their future learning or work (Bell 2010). Several experiments also have demonstrated that learners can acquire different kinds of knowledge and skills when using PBL approach (Boaler 1999; Geier and Blumenfeld 2008; Thomas 2000). Norman and Schmidt (1992) identified four powerful positive strengths of PBL:

1. Learners' learning motivations will increase.
2. Learners will become problem solvers and be more self-directed.
3. Learners will learn and memorize information better.

4. Learners will apply their obtained knowledge and skills to solve the new and challenging problems.

Despite these advantages, a number of problems during the PBL implementation have been reported, which result in the failures of PBL in educational practices. In order to discover the hindering factors of PBL, English (2013) interviewed 182 tutors. In his study, “*student struggles*” was identified as the major resistance factor, which included learners’ low academic abilities, low motivations, ill behaviors, poor self-directed or self-regulation skills, inequality in group work contributions, and the resistance to PBL. Helic and Krottmaier (2005) also discovered that “*learners usually had no problems in generating project plans but they had problems in following the generated project plan and managing their time*”. When we look back into the conventional learning to explore the reasons behind, it is not hard to find that most learners are accustomed to passive learning, in which tutors give lectures and the learners only take notes in a passive manner. English (2013) also pointed out that the transition from tutor-centered to student-centered methods should be done gradually, with appropriate modeling, scaffolds, formative assessments, and feedbacks.

Hence, in order to be successful in PBL, we have to support learners to develop self-regulation skills. Learners need to take responsibility for their learning processes by setting goals, monitoring, reflecting, and sustaining their motivations from the beginning of the project until the end (English and Kitsantas 2013). These skills are the main abilities of SRL. Wolters (2003) affirmed that “*self-regulated learners are thought to hold a collection of adaptive beliefs and attitudes that drive their willingness to engage in and persist at academic tasks*”. Gehlbach and Roeser (2002) also asserted “*the more that students perceive autonomy, the more engaged they become in learning*”. Therefore, in the next section, we study the SRL process by considering the definition, models and characteristics.

2.1.2 Definition of SRL and SRL tools

Pintrich and Zusho (2007) defined Self-Regulated Learning as “*an active constructive process whereby learners set goals for their learning and monitor, regulate, and control their cognition, motivation, and behavior, guided and constrained by their goals and the contextual features of the environment*”. Pintrich (2000) proposed a SRL model by classifying the different processes composing of SRL (see Table 2-1). There are four phases of this regulatory process: planning, self-monitoring, control and reflection, which are described from four regulation areas: cognition, motivation/affect, behavior and context.

1. **Planning.** Learners are required to set their target goals or the specific objectives (time and resources distribution); identify their prior knowledge applied for the project and active metacognitive knowledge (self-efficacy, identify the difficulties of each task, recognize the gaps between the reality and the goals, distinguish the recourses and knowledge which are helpful to

- task delivery etc.); plan and design the methods to observe their own behaviors; generate the perception of project and context.
2. **Monitoring.** Learners need to be aware of their cognitions (Do I understand this knowledge? What other related knowledge need to learn? Have I reached the goals I set?), motivations and affective states (Am I upset or confident about the learning? Can I complete it successfully?), effort and time usage (Do I spend enough time in the learning? Do I try my best to learn? Do I need help?), and task and context conditions (What is the progress of the learning? What other learning recourses I need?).
 3. **Control.** This phase includes the selection and adaptation of appropriate strategies for cognition, motivation and emotion, increasing or decreasing of efforts or time, deciding to persist or give up learning, seeking other helps, changing or leaving the tasks and the context.
 4. **Reaction and reflection.** Learners need to judge or evaluate their project outcomes compared with the goals set in the first phase, reflect their own behaviors in the whole processes, sum up their shortcomings and achievements that can be avoided and promoted in the future work.

Table 2-1 Pintrich's SRL model (Pintrich 2000)

Phases	Regulation areas			
	Cognition	Motivation/Affect	Behavior	Context
Fore-thought planning, and activation	Target goal setting Prior content knowledge activation Metacognitive knowledge activation	Goal orientation adoption Efficacy judgments Ease of learning judgments (EOLs); perceptions of task difficulty Task value activation Interest activation	Time and effort planning Planning for self-observations of behavior	Perceptions of task Perceptions of context
Monitoring	Metacognitive awareness and monitoring of cognition (Feeling-of-Knowing, Judgment of Learning)	Awareness and monitoring of motivation and affect	Awareness and monitoring of effort, time use, need for help Self-observation of behavior	Monitoring changing task and context conditions
Control	Selection and adaptation of cognitive strategies for learning, thinking	Selection and adaptation of strategies for managing motivation and affect	Increase/ decrease effort Persist, give up Help-seeking behavior	Change or renegotiate task Change or leave context
Reaction and reflection	Cognitive judgments Attributions	Affective reactions Attributions	Choice behavior	Evaluation of task Evaluation of context

Zimmerman (2002) proposed a three-cyclical-phases structure of self-regulatory processes: forethought, performance and self-reflection. In the forethought phase, it includes task analysis (goal setting and strategic planning) and self-motivation beliefs (self-efficacy, outcome expectations, intrinsic interest/value and goal orientation). In the performance phase, it includes

self-control (self-instruction, imagery, attention focusing and task strategies) and self-observation (self-recording and self experimentation). The self-reflection phase includes self-judgment (self-evaluation and casual attribution) and self-reaction (self-satisfaction/affect and adaptive-defensive responses).

From these two SRL models, we can find that in order to help learners to self-regulate, we need to help them to follow and perform the phases mentioned above in a learner-centered learning environment.

Regarding to the methods that can support these phases, Cleary and Zimmerman (2004) proposed a micro-analytic assessment questionnaire (see Table 2-2) based on the above three-cyclical-phases. This questionnaire can supply the specified guidance on setting goals, monitoring the effectiveness of strategy, self-evaluation, and adjusting strategies when it is not effectively. This questionnaire use the math test as an example and it can be modified to adapt to any subject area. After learners answer the assessment questions, they can generate a cyclical thought and actions (for example, My goal is . . . and will be reached by . . . ; I got grades of . . . by using the strategy . . . ; I did not attain the goal . . . in my last test; The strategy I used . . . was ineffective; I need to modify to . . .).

Table 2-2 The micro-analytic assessment questions (Cleary and Zimmerman 2004)

Phases of cyclical feedback loop	Self-regulation processes	Assessment questions
Forethought	Goal Setting	Do you have a goal when studying for your math tests? Explain.
		Do have a goal you are trying to achieve on your math tests? Explain.
	Strategy Choice	How did you decide to use this strategy when preparing for math tests?
	Self-Efficacy	How sure are you that you can get an 85 on your next math test?
		How sure are you that you can solve 70% of these math problems?
	Intrinsic Interest	How interesting is studying/preparing for your math tests?
How much do you enjoy studying/preparing for your math tests?		
Performance	Attention Focusing	Do you have to try to motivate yourself when studying for math tests?
		What do you do when you don't feel like studying for your math tests?
	Self-Recording	Do you keep track of where you study for your math tests?
		Do you keep track of how long you study for your math tests?
Self-reflection	Self-Evaluation	How do you determine if you performed well on your math test?
	Satisfaction	How satisfied are you with your performance on your last math test?
	Causal Attributions	What is the main reason why you got a 75 on your last math test?
	Adaptive Inferences	What do you need to do to improve your performance on your next math test?

However, Hadwin and Winne (2001) found that learners would vary their responses when they were required to respond of different tasks within a course. Winne and Jamieson-Noel (2002) concluded that the responses are ambiguous because learners may distort or inaccurately report the qualities of their SRL, unless some measures can reveal what is the course contexts in learners minds when they are answering the questions.

Feedback is another way to help learners to self-regulate. Nicol and Macfarlane-Dick (2006) defined feedback as “*information about how the student's present state (of learning and performance) relates to these goals and standards*”. Butler and Winne (1995) asserted that feedback is an inherent catalyst of SRL, which can generate information on learning performances and accelerate learning. The feedback can be classified into two categories: internal and external. Internal feedback is generated when learners monitor their learning behaviors and current progresses and compare their achievements with the desired goals. They use this information to determine whether they can continue as before or they need to change their ways of carrying out the projects. Meanwhile, the external feedback, coming from tutors' remarks or peers' contributions or computers' tracking, also supplies complementary information for learners to regulate learning. However, most learners are passive learners and they are willing to obtain the external feedback rather than to generate internal feedback. Nevertheless, internal feedback is critical in shaping the evolving pattern of a learner's engagement with a task (Butler and Winne 1995). Hence, it is necessary to help learners to monitor their learning behaviors, reflect their learning progresses and judge their learning achievements in order to generate internal feedbacks. Nicol and Macfarlane-Dick (2006) proposed seven principle of feedback of SRL:

1. *helps clarify what good performance is (goals, criteria, expected standards),*
2. *facilitates the development of self-assessment (reflection) in learning,*
3. *delivers high quality information to students about their learning,*
4. *encourages teacher and peer dialogue around learning,*
5. *encourages positive motivational beliefs and self-esteem,*
6. *provides opportunities to close the gap between current and desired performance,*
7. *provides information to teachers that can be used to help shape the teaching.”*

In SRL, students are no longer traditional learners but metacognitive and active learners. They need to set goals, organize a set of actions, select adaptable strategies, self-monitor and self-evaluate their behaviors and learning (English and Kitsantas 2013). According to Paris and Winograd (1999), self-regulated learners can be characterized by: sustaining motivations; applying suitable strategies; reflecting their own behaviors; setting attainable, appropriate and challenging goals; and managing time and resources. However, Zimmerman, Bonner, and Kovach (1996) has pointed out that “*there is seldom any instruction in methods of studying or other self regulatory skills, and there is substantial*

evidence that many students fail to acquire these skills on their own". He (Zimmerman 1998) also emphasized that even though "*academic self-regulation and its constituent forms of self-reflection are seldom taught in most schools*", students still can obtain these SRL skills through "*a core set of instructional and personal practice experiences*". Pintrich (1995) proposed that self-regulation can be improved by developing learning environments which make learning processes explicit by meta-cognitive training, self-monitoring and self-regulation practices.

Based on the above statements, an important and meaningful issue in PBL is to define how to develop an "*instructional and personal*" learning environment to help learners to self-regulate. In the next section, we analyze the learning environments that support SRL in PBL.

2.2 E-learning systems used to support SRL in PBL

With the development of network technology, E-learning has become popular in delivering training and instruction for all academic levels worldwide. Many E-learning systems have been developed to support PBL. In the E-learning systems, learners can access multimedia (for example, texts, images and videos) courses' contents, as well as interact with tutors and the other course members through messages, forums, video-conferences, chats or the other types of communication tools (Sánchez and Hueros 2010). Large amounts of researches have proved that E-learning system can improve PBL experiences and effectiveness. Brodie and Gibbings (2007) examined the use of an E-learning system for learners to undertake a PBL course in the University of Southern Queensland. Learners used the platform to carry out the group works, such as team meetings, team communications, team electronic submissions and individual assessments. The authors found that the E-learning system allows learners to appreciate the value of participation, trust and mutual respect, encourages reflective thought and dialogue with others, and validates new ideas to interpret learning experiences. Clark and Mulligan (2011) carried out an experiment with 13 learners to participate in two sessions of nine weeks long PBL courses (one on oral medicine and one on special patient care). Most of the participants agreed that the various materials contained on the E-learning system were helpful and half of the participants agreed all future PBL courses should be supported by a web-based learning system. Sulaiman (2013) explored learners' acceptances of PBL online systems in terms of learning outcomes (students' knowledge, skills and application of knowledge and skills; communication; and independent learning) in a physics course of University Malaysia Sabah. He found that the E-learning system could train learners to be more competent in communication and more independent, improve learners' skills especially in handling their own learning activities and resource findings. The advantages of E-learning system in PBL can be concluded as follows:

1. ***E-learning system provides a rich and functional learning environment for PBL.*** It offers learners a variety of instructional and communication tools, such as course content delivery tools, synchronous and asynchronous conferencing systems, polling and quiz modules, virtual workspaces for sharing resources, white boards, grade reporting systems, logbooks, assignment submission components, etc (Za'ane 2002). Furthermore, E-learning system integrates these tools together to decrease learners' intellectual workloads comparing with using numerous software products separately. Brodie and Gibbings (2007) surveyed a large number of learners and found that the E-learning system helped them to manage the project activities because it provides a solid structure and offers a wide range of important tools for effective distance collaborative PBL.
2. ***E-learning system provides a comfortable and "community sense" environment for PBL.*** In spite of unable face to face meeting, distance learners still can feel a great sense of "community" in E-learning systems, which fosters collaborative learning in PBL (Brodie and Gibbings 2007). E-learning system creates a virtual community for learners and helps them to carry out the project activities, including organizing project meetings, team communications, documents sharing and peer-assessments, in which they can have a sense of involved and the group cohesion can be enhanced.

Helic and Krottmaier (2005) proposed four facilities that E-learning system should incorporate to support PBL:

1. ***Support for project management.*** Tutors or learners should be able to define the project plans in the E-learning systems, which consist of a set of tasks for learners to complete. The project plan incorporates a time schedule that sets a period for accomplishing each particular project tasks.
2. ***Learners are central to the learning process.*** E-learning system need to support learners in many different ways to achieve project goals by supplying simply functions. The project needs to be designed based on learners' knowledge levels and suit to their abilities. E-learning system should not restrict the technical environment or the file format.
3. ***Support for learners' collaborations.*** A teamwork environment should be created to support the collaborations between learners and tutors, e.g. a common area to write documents, modeling, upload/download files.
4. ***Support for data analysis.*** E-learning system is required to assess learners' performances and provide the valuable feedbacks to learners or tutors. In PBL, it is essential to keep learners on the right way to solve problems and guarantee tutor to intervene at the right time.

Next, we analyze several E-learning systems used in PBL according to the above four requirements proposed by Helic. These systems seem relevant to our context because they not only support learners to achieve projects collaboratively, but also aim to improve learners' self-regulation skills.

The SPLACH system (George and Leroux 2001, 2002) aims to help young learners to design and model micro-robots and their driving systems. It

consists of asynchronous communication tools (E-mail and discussion forum), a synchronous meeting tool allowing learners to collaborate smoothly, a calendar scheduling tool to help learners to manage the progresses of the project, a tool to write reports during the project and the specific educational robotics tools which are student-centered to supply special project knowledge. Based on the theory of Conversation and Communicative Acts, the system can calculate the learners' behavior profiles (moderator, seeker, independent and valuator) based on the communicative acts selected by them. The authors divided the acts into five categories: initiative acts (suggesting, asking, affirming), reactive acts (answering, querying) and appraising acts (approving, disapproving), greeting acts (greeting), auto-reactive acts (clarifying, correcting). According to learners' behavior profiles, the tutors (or the learners themselves) could gain insight into learners' (their own) behaviors. However, in this case, only the communicative acts are calculated rather than learners' communication contents, which are also particularly important for the system to analyze in order to observe learners' reflections. Furthermore, it is not sufficient to consider merely the communication activities. Communication is one of the most important aspects during the collaboration but it cannot represent all. We think that the E-learning systems have to supply the feedbacks covering all aspects of project activities.

Laffey *et al.* (1998) proposed an E-learning system which integrates a set of internet-based tools to provide intelligent supports for the project. A shared dynamic knowledge base is also created to support working and learning in a PBL community. The architecture of the system (Table 2-3) consists of two instructional processes (scaffolding and coaching) and four learning processes (planning and resourcefulness, knowledge representation, communication and collaboration, and reflection). It supports reflection by requiring learners to articulate their works, by facilitating comments and critiques from others, and by making it easy to review and compare present work with previous work. With the help of several discourse and collaboration channels, it increases the likelihood that the learners' articulations will be reviewed and critiqued, thus it encourages further reflections. The system tracks and stores each revision made to project's documents from the beginning of a project to its end. Even though this system can help learners to reflect on their understandings of their works, it is still insufficient for learners to self-regulate their project and learning. There are no external feedbacks for learners supplied by the system to stimulate their deep reflections. In fact, what learners really want to know is whether their performances are better (worse) than the others' and what the reasons are, whether they are on the right way to work, whether they have reached the goals set in the project plan etc. We think an E-learning system needs to give a valuable and detail feedbacks to learners to enable them to control their projects and have an insight of their performances, rather than just supplying the differences between present work and previous work.

Table 2-3 The architecture proposed by Laffey (Laffey *et al.* 1998)

Processes	Definitions	Methods
Instructional		
Scaffolding	Structural supports to assist novice learners in the performance of tasks for which they would otherwise be unprepared.	Interface design broadly scaffolds the steps of a project, the language of real science, and concerns which must be addressed in order for a project to be successful.
Coaching	Situated responses to learner task performances which are targeted at bringing learner performance closer to expert performance.	Advanced, interactive help system that is context/task sensitive. Immediate feedback targeted at improving the use of the tools themselves. Immediate feedback targeted at explaining/scaffolding/ supporting performance at various project tasks. Context sensitive guidance system.
Learning		
Planning & Resourcefulness	Tools designed to assist learners with the complex demands involved in planning and being resourceful within authentic research projects.	Scheduling tools for establishing specific objectives and their start and stop dates. Resources tool for specifying material and information resources necessary for the project, with linking to specific objectives. Team member/ member responsibility specification tool.
Knowledge representation	Tools designed to assist learners in the framing, representation, and re-representation of their ideas, knowledge, and their development, and in deriving cognitive benefits from the act of representation.	Sections for representation of a project abstract, project goals, objectives, resources, and applications/extensions of the work. Multiple representational formats via native documents and automatically generated/uploaded WWW pages. Scaffolding, coaching, and guidance systems fully integrated to assist in the representation process.
Communication & Collaboration	Tools designed to support the exchange and sharing of ideas and results, collaboration between widely distributed participants, feedback, discussion, & debate, and the growth of a "community" of learners.	World-Wide-Web based comment forms. Site customizable, threaded, public and private discussion groups with embeddable URL's for resource sharing. Integrated email with address book and embeddable URL's. Integrated point-to-point and group real-time chat facilities
Reflection	Tools to support self and communal evaluation and reification of previously completed work, with subsequent cognitive and physical revision, reframing, and restructuring of ideas, assumptions and representations.	Tracking and storage of all revisions to a team's work. Multiple-window views for comparison of old and new work. Sharing of all work, including old and new revisions, with the larger community. Scaffolding, coaching, and guidance systems fully integrated to assist in the reflection process.

Helic and Krottmaier (2005) proposed a learning system in PBL in a web environment. This system integrates three components: a special document, a special discussion folder, the collaboration and communication tools and an

evaluation tool. The document is composed of the course and project motivation, problems that need to be solved, goals, etc. The folder is provided to present a sample project with the project plan definition, a number of project discussion and all learners' contributions. The tools are used to evaluate learners' works in order to provide them valuable feedbacks and further directions. In this case, tutors give direct feedbacks to learners. Strictly speaking, it is reflection, not self-reflection. Tutors discover the learners' problems during the project and give them feedbacks into understandable words. Learners are always the information receptors, not the explorers.

From the above examples, we can find that PBL can engage the learners in activities that simulate the demands of real life professional practice and E-learning system can greatly aid the learners to perform the PBL without the restraints of the time and locations. The E-learning system supplies learners with a platform that supports online learning, long-term learning content retention, group cohesion, shared knowledge, communication skills and collaborative work experience. However, compared with the requirements proposed by Helic, we can find the above E-learning systems can support learners' collaboration and project management while they are failing in supplying valuable data analysis to learners. E-learning system merely help learners to generate lower-level skills such as "memorizing", "understanding" and "applying" rather than high-level skills practices, like "analyzing", "evaluating" and "reflection" (George and Leroux 2002). Ku and Chang (2011) affirmed that the existing E-learning system could not improve learners higher level think, or support active learning attitude. George and Leroux (2001) pointed an effort has to be made in E-learning system to help learners to reach these high-level skills. By analyzing the above systems, we conclude two points that need to be taken into account when we design an E-learning system for PBL to provide valuable feedbacks:

1. ***All the project activities have to be considered.*** In a context of the project, learners have to carry out a range of activities. In order to supply a general view on learners' performances, it is necessary to provide the feedbacks based on all the activities and supply the further data analysis. We think that a helpful feedback has to supply the information about learners' activities (for example, time spent on the activities), learners' social states in the group (for example, the learner is active or inactive in the group), learners' cognition progresses (for example, learners' progresses in writing a document or gaining a skill) and learners' states of minds during the project (for example, learners' affective states).
2. ***The feedbacks have to be offered directly to learners.*** We think that it can help learners to generate self-reflection if the feedbacks are offered to learners themselves directly rather than tutors transform and interpret the feedbacks to learners.

From the above statements, we can find that it is necessary for the E-learning systems to support self-regulation in PBL by providing learners with the feedbacks from a global point of view. We are interested in the methods, which

can explore the traces left in the E-learning systems when learners carry out the projects. Traces can reflect how learners work together in a project from the aspects of planning, evaluating, and problem solving and describe the observable interactions between learners and project contexts accurately and chronologically. By analyzing the traces, learners can improve their performances and obtain the high-level skills to self-regulate their behaviors, without relying on poor assessments and distort memories in questionnaires we introduced before. In the next section, we analyze the traces used in E-learning systems and do a comparative analysis between different types of traces.

2.3 Traces and tools used to support SRL in PBL

In order to support learners to reflect or self-regulate in PBL, the E-learning systems should guide learners to carry out the projects and offer a critical feedback about their behaviors and learning. One of the methods can meet this requirement by analyzing the traces left by learners when they operate the E-learning systems. Traces can not only help learners to have an insight about the way how they select, monitor, assemble, rehearse, and translate information to learn, but also provide raw information for mapping SRL and its effects (Perry and Winne 2006). The conception of trace is defined as “*a collection of temporally situated observed elements*” while the observed element is defined as “*any structured information resulting from observing interactions*” (Djouad and Mille 2010). These traces can be calculated and be presented in the form of indicators. The indicator is referred to a piece of information presented in a visual form, to reveal the mode or the process of the activity, the characteristics and the quality of the interaction product, or the mode, the process, and the quality of the interaction being performed on an E-learning system (Dimitrakopoulou, Petrou, and Martinez 2006).

In order to analyze how the traces are used, we define two types of traces. The first is **activity trace**, which is defined as “*the users' actions recorded directly by the E-learning system during the learning activities*”. Another type of trace is **reporting trace**, which is “*reported by the learners themselves to explain how their (or their peers') activities out of the E-learning system are carried out*”. For example, the descriptions of the activities accomplished out of the E-learning system and the activities results, such as the contents of face-to-face discussions.

2.3.1 Activity traces in learning environments

ESSAIM (Després C. 2002) is a tool to help tutors to obtain the information about how learners carry out the project. It presents the learners' progresses of activities from a global view. Tutors can get different information about the projects, for example, activity progresses, time spent in the project and the interaction degrees between learners and computers. FORMID (Gueraud and

Cagnat 2004) has been developed to help tutors to monitor learners' synchronous individual and group project activities, which are presented in a Gantt from three levels: a global view of the group progress, a detail view of the group progress in a given step and a chronological view of a learner progress in a step. These two tools explored activity traces and have been designed mainly for tutors. No function has been designed for learners to enable them to manage their own learning activities during long periods.

Croisière (Teutsch, Bourdet, and Gueye 2004) and Reflect (Despres and Coffinet 2004) have been developed to monitor asynchronous learning activities and help learners to become autonomic by regulating their learning activities. Croisière allows learners to choose the learning activities according to their own learning strategies. Reflect presents the progresses of a learner or a class and supply the feedbacks about it. Learners plan their activities done in the course and tutors decide to accept their plans or not according to the amounts of tasks they have finished. TACSI (Laperrousz, Leroux, and Teutsch 2006) presents learners' activity traces collected in the collective projects, such as their contributions in the collective activities or discussions and their social behaviors. CourseVis (Mazza and Dimitrova 2007) use activity traces produce by WebCT in order to compute graphical indicators about learning behaviors, social characteristics and cognitive evolutions of distance students. TrAVis (May, George, and Prévôt 2011) enables users to directly access to tracking data repository, by a Graphical User Interface, in order to compute indicators and choose the visualization modes. It is a reflective tool giving learners information about the way they carry out discussions or other collaborative activities.

From studying these E-learning system using activity traces, we can find that the data sources supporting self-regulation come from log files, which are captured by computers automatically. Because learners cannot access to the trace collection process, learners have no possibility to complete the traces by other information (such as their thoughts, emotions, activities done outside of the E-learning system, etc.). However, for most projects, learners also need the help of the other computer software (Skype, YouTube, Firefox, MS office etc.), probably need to do some outdoors activities (visit factories, face to face discuss, collect temples for testing etc.). They are also required to set their project plans, assess themselves and their peers, etc. Unfortunately, these data are not collected and analyzed by the E-learning systems.

2.3.2 Reporting traces in learning environments

Blogging and electronic portfolio provides learners with a place to record the information related to their learning manually and support them to adjust their learning processes through reflection. The information recorded by learners composes the reporting traces, which can be analyzed and explored. Blogging has been increasingly recognized as a popular web technology for education, especially in distance learning settings. Blogs are designed to allow the simple and fast creation of web contents by using publication functions, such as posts,

comments, and instant hyperlinks to information sources (Du and Wagner 2007). Blogging has been recommended as a suitable tool for learning. Chu, Chan, and Tiwari (2012) examined the blogging effects in the internships among the undergraduate students majoring in information management and nursing in the University of Hong Kong. He found that the blogging is helpful to construct the knowledge, solve the problems, self-reflect and communicate emotions in the internships. EnquiryBlogger (Ferguson, Shum, and Crick 2011), a WordPress plugin, proposes a set of indicators related to the use of blogs created for learning. Learners can write new blogs and categorize them according to seven dimensions related to “learning power” (changing and learning, creativity, critical curiosity, learning relationships, meaning making, resilience and strategic awareness), eight dimensions related to the “enquiry dynamics” (personal choice: concrete place/object, observation-description, generating questions, uncovering narratives, mapping, connecting with existing knowledge, interface with curriculum requirements, assessment-validation), and their states of minds. By analyzing these reported data, EnquiryBlogger can generate 3 indicators providing visual feedbacks to learners: a graphical representation of the enquiry dynamics assigned by learners when they post blogs; a spider diagram presenting the progresses of learners’ learning powers self-assessed by learners themselves; a line chart reflecting learners’ affective recorded by learners when they write a blog.

An electronic portfolio (EP) is defined by Abrami and Barrett (2005) as “*a digital container capable of storing visual and auditory content including text, images, video and sound*”. EP can help learners to organize the learning contents, support the pedagogical processes and assessments, and share learners’ works with the others and gain feedbacks. Glaser-Zikuda and Fendler (2011) indicated that the core issues of the portfolio approach are the processes of planning, documenting, and reflecting on individual learning activities. EP enables learners to improve the skills of metacognition and self-regulation. It helps learners to set goals and plan, monitor and reflect on learning, and assess their performances with the goals.

After we analyzed these researches, we find that it is helpful for learners to write blogs or e-portfolios but it is difficult for them to focus on the subjects related to learning and activities. These tools supply a space for learners to write their learning progresses and learners can write anything they want freely, even maybe the contents are irrelevant to their learning. Moreover, there is no guide defining how to organize the content to be more effective to support reflection and learning processes. Therefore, it is difficult both for learners to record correctly and for tutor to give valuable feedbacks. Finally, most blogs and e-portfolios are non-structured texts and the contents are hard to be analyzed automatically by the E-learning systems. Even though, Natural Language Processes (NLP) can be used to analyze the contents and help learners to discover specific information or relations between them. Nevertheless, language analysis cannot be completely automated and so produced data cannot be

continuously grasped with activity traces to produce indicators usable during a learning session or project.

Pco-Vision (Michel, Lavoué, and Pietrac 2012) is applied in PBL by using the structured self-reporting. It provides learners with a global view on objectives-actions-results in order to support self-regulation and to develop complex abilities (e.g. evolution ability). Pco-Vision supports the plan and judgment processes of PBL. The traces are self-declared and are presented from an individual view and a collective view. Pco-Vision supplies four indicators, which present (1) the knowledge levels that learners have acquired in comparison with the target levels that they have to acquire defined in the project plan, (2) the trends of their states of minds (morale and satisfaction) during the last four weeks, (3) the tasks to do, presenting in the form of a post-it note, (4) the key events noted by learners (like in a blog).

Nevertheless, the reporting process of Pco-Vision, organized as self-declaration only about the current week, is considered by learners to be too much bindings (Michel *et al.* 2012). The learners are required to answer the questions proposed by the designers and select one of the prepared answers. They neither can write their own answers nor write what they want to write in the reports. The designers of Pco-Vision (Michel *et al.* 2012) pointed out the need of contextualized data declaration interfaces as well. Furthermore, all the indicators are calculated based on reporting traces totally, which can reflect learners' thinking and self-judgments. However, the way of how learners carry out the project is ignored. Learners are unable to monitor their behaviors and performances resulting in the doubts such as "*Am I in the right way to work? Why do I fail in this task?*"

From the above researches, we can find that the reporting traces can help learners to collect more information related to the project that cannot be captured by the systems, such as learners' states of minds, learning experiences etc. Blogging and e-portfolio supply a non-structured input method and allow learners to input information freely, but there is no guide to help learners to write the information relating to the project. Furthermore, it is difficult to analyze the natural language for the system. Considering the structured input method proposed in Pco-Vision, we think it is too inflexible and constrained for learners. It is impossible to record information totally based on the learners' situations. Hence, it is important to find a flexible method to enable learners to record information based on their own situations and the system can capture the information for analysis automatically.

2.3.3 Hybrid traces in learning environments

The Learning Kit Project (Winne, Hadwin, and Gress 2010) is a collaborative learning platform proposing different tools of communication and collaborative writing (gStudy, the Coach, gChat, and the LogAnalyzer) to support distance learning. gStudy collects detailed, time stamped trace data, such as selecting the information to process, highlighting the learning contents, choosing the options

in menus, clicking a button, creating links, opening and closing windows etc (Winne 2006). The LogAnalyser tool analyzes these activity traces and the results are presented in indicators ranging from simple frequency counts to graph theory statistics (Perry and Winne 2006), such as the frequencies of events, properties of event (length, duration, density of information) and properties of patterns of learning events. Chat logs can provide the content of a dialogue (idea exchange and generation, the concepts proposed in the discussion) and the context of a dialogue (collaborative scaffolds usage: roles and prompts). Analyzing chat logs provides opportunities to trace learners' discourse patterns and idea transitions when learners attempt to co-construct shared meanings, plans, and reflections of a task (Winne *et al.* 2010).

In the case of gStudy, activity traces (analyzed by LogAnalyzer) and reporting traces (Chat logs) are used to support self-regulation in learning. LogAnalyzer can analyze the activity traces and support tutors, researchers and learners to know more about how learners learn and self-regulate collaborative learning. Although gStudy can present the chat logs to stimulate learners to generate reflections on learning, in fact, the level of usage of chat logs are not very high because the system cannot analyze the contents of the chats. Furthermore, activity traces and reporting traces are not really integrated together. Two types of traces work separately because the activity traces do not cross with the reporting traces to provide more meaningful and more precise analysis. For example, in the chat logs, a learner plans his project schedule. Then, LogAnalyser captures the activity traces of the project activities done in real. The system would combine this information together so that this learner can know whether he/she follows his/her schedule or not.

“MIRROR Integrated User Profile” application (MUP App) (Fessler, Wesiak, and Luzhnica 2014) is developed to integrate, synthesize, analyze, and visualize traces captured by several different applications in order to arouse and support higher-level reflection possibilities. It integrates two different applications, named KnowSelf and the MoodMap. The KnowSelf application, targeted towards learners who work mainly on a PC, is intended to support reflection on time management and self-organization. It visualizes the time that learners spend on the computer, the software and resources. The interactions (applications and resources usages, window focuses and titles, focus switches) between learners and PC are captured automatically and are presented from a general view by different visualizations modes. Additionally, the learners can manually record their reflection diaries after they have a look at the statistics provided by KnowSelf. The MoodMap allows learners to declare and track their states of minds during a working day by simply clicking on a bidimensional colored map. Each mood is composed of two dimensions, namely valence (negative to positive feelings) and arousal (low to high energy). It provides learners with trends charts to enable them to reflect on their moods development over time or with comparison charts of one's own mood with the moods of others. Learners can also insert the reasons to explain their moods further.

In this case, we can find that activity traces and reporting traces are explored to provide higher-level reflection feedbacks. The authors found that combining data collected from different applications, analyzing and visualizing them together can further promote reflective learning and enhance awareness of the work life (Fessler *et al.* 2014). Nevertheless, the activity traces are produced in the computer environment while the reporting traces are related to learners' states of minds. In other words, the tasks or activities done out of the computer and the other types of reporting traces (e.g. self-assessments, discussion contents) cannot be captured. How to collect the traces comprehensively to reflect learners' project progresses completely is a critical issue. Secondly, its personalization level is weak. Learners only can customize the visualization of indicators (time span and the filters on user, chart etc.) rather than for the processes of traces treatments. Therefore, learners cannot structure their own traces in order to build personalized indicators and so their levels of involvement are limited.

2.3.4 Discussion

In this section, we have analyzed the usage of two types of traces in the E-learning systems to support self-regulation in PBL: activity traces and reporting traces. After study the different traces used to support self-regulation, we can find that both the activity traces and reporting traces have their own strengths.

Many E-learning systems collect the activity traces and explore them to support SRL because they can be captured by the systems in the background and they are more structured than the reporting traces and easier to be analyzed. Furthermore, systems can capture these activity traces automatically when learners interact with them, without interrupting the interactions. The E-learning systems can record the interactions between learners and the systems' server sides (login/logout, open/edit/delete files, click button, navigate between pages etc.) and between learners and learners by computers (chat with others, comment others' forum posts, send private messages to others, join video conference etc.). All of these interactions supply abundant data for analyzing in order to provide learners with valuable indicators to self-regulate. Most quantitative indicators are produced by activity traces: the time spent on a specific task or resource, the interaction frequencies between two or more learners, the usage frequency of a specific tool etc. Based on these indicators, tutors can have an insight of how learners carry out the projects and learners can observe their weaknesses in the collaboration work so as to self-regulate their behaviors.

Reporting traces give the possibility to learners to involve in trace collection. Learners can add the information, which cannot be captured by the E-learning systems, into data analysis. The traces of external activities (done out of the E-learning systems or out of the computer environment, e.g. create a new file by MS Word, send email by Gmail, search information by Firefox, visit factories etc.) and non-instrumented activities (accomplished out of the computer environments without instruments, e.g. face to face discussion, brainstorm,

judgment, set goals etc.) can be reported manually by learners. Reporting traces written by learners manually carry more semantic information than activity traces. Learners can insert their thinkings, the states of minds, explanations, reflections into reporting traces. The indicators based on reporting traces are fundamental because the contents are based-text and are hard to be analyzed automatically by the systems. Most systems merely list the raw reporting traces without treatments and the tutors have to do the analysis works.

Both the activity traces and reporting traces are significant in PBL to support self-regulation (SR). Regarding to the above researches of the two types of the traces, we assert that it is limited to use them separately. We hypothesize that integrating and analyzing activity traces and reporting traces together can support SR in PBL greatly. We hope to achieve a high level analysis of the project activities in a global way to help learners to improve their learning performances and self-regulation skills. However, another question has to be considered: how to help learners to insert relevant and effective reporting traces of their projects? Due to the different ages, genders, education backgrounds of the learners, reporting traces are different from learners to learners. If learners insert the information unrelated to their projects and learning, it can not only increase learners' workloads, but also can hinder the traces analysis even lead to wrong results, for example, the system cannot capture the relevant information or capture the wrong information from the reporting traces. Furthermore, the second question is more challenging: how to integrate activity traces and reporting traces? Because activity traces are structured and the reporting traces are not, it is difficult to integrate and process the two inhomogeneous types of traces. The design of an uniform data model to store all the traces is an issue needed to be considered.

The third question is related to the presentation: how to present these traces to learners from a more understandable and explicit angle. It has been confirmed that dashboards can support learners' self-judgment processes, foster insights, increase self-control and promote positive behaviors, especially when dashboards present the information about how the activities are carried out (Ji *et al.* 2013). The use of a dashboard is a key-learning objective in the context of PBL by providing review and analysis of their personal history on the plenty forms of visualization modes, e.g. bar chart, bubble charts, social network, timeline chart etc. These visualizations provide broad insights on learner activities (Charleer, Odriozola, and Luis 2014). In the next section, we focus on the different types of dashboards for trace presentation.

2.4 Trace presentation: simple dashboard and dynamic dashboard

A dashboard used in PBL should present the 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). The

dashboard could be considered as a container of indicators (Ji 2012). Experiments show that the dashboard has an impact on learners' grades and retention knowledge (Verbert, Duval, and Klerkx 2013). The dashboard is a good way to strengthen learners' self-regulation skills and to facilitate the group works, the group cohesion and the professionalism of learners by showing explicitly the consequences of their acts (Michel *et al.* 2012). Furthermore, the visualization of these traces in the dashboards can assist the learners in creating a feedback loop of awareness, reflection and sense-making (Verbert *et al.* 2013) and so improve their motivation (Santos, Charleer, and Parra 2013). Michel *et al.* (2012) argued that “*the awareness of the project goals compared to the results of the activities will help the self-monitoring of the skills applied to achieve the project*”.

Regarding the existing researches, we classified the dashboard into two categories based on the customization levels: simple dashboard and dynamic dashboard. Simple dashboards enable the users to observe pre-defined indicators (all the aspects of the indicator are defined by the system developers totally) and dynamic dashboards let users customize their own indicators (users can define all the aspects of an indicator). In these cases, users include learners, tutors or researchers. In our context, a dynamic dashboard is composed of customizable indicators. We define three levels of a customizable indicator: low level supports users to set some simple parameters to filter the results of the indicator, medium level allows users to define the calculation functions and change the visualizations of the indicator, 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. Next, we analyze several simple dashboards and dynamic dashboards.

2.4.1 Simple dashboards with predefined indicators

Study desk (Narciss *et al.* 2007) is a working space for learning and studying, which is designed to support different learning contexts, for example, lectures, seminars or project-based courses. Tutors can upload multiple materials and information of a specific topic. Thus, learners not only are required to prepare for courses or projects, but also need to repeat and elaborate knowledge in a self-regulated way. It supports to monitor learners' learning processes by offering access to the protocol of all learning activities (progress and task report), which aims to help tutor and learners to discover/correct errors, find knowledge gaps, overcome obstacles and apply more effective strategies towards solving learning tasks. The system will present the indicators to learners (see Figure 2-1), which can (1) present the percentage of correctly solved tasks, partly correct solved tasks, incorrectly solved tasks and further tasks in a bar chart, and (2) give an evaluation of the overall performance. From these indicators, learners can know which tasks they have already completed, the quantity of materials and recourses still at their handlings and the percentage of the completed and uncompleted learning tasks.

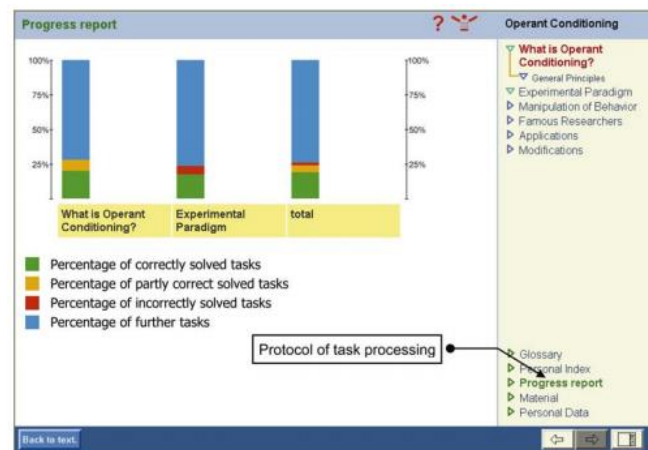


Figure 2-1 The dashboard of Study desk (Narciss *et al.* 2007)

Feeler (Durall and Toikkanen 2013) is a paper-designed dashboard that supplies statistical analysis and visualization of the relationships between learners' learning performances and their well-beings in order to improve reflection and self-regulation awareness. It integrates the personal physical information (such as the amount of physical activities and concentration levels) with Learning Analytics to help learners to reflect on their concentration levels and physical activities during the learning. Thanks to a headband and smart textiles, different mental activities can be registered to determine a person's level of concentration. The mental data are transferred to a wool wristband, in which the small led lights will blink when the learner loses attention for a period of time. The indicators are presented together through a screen in the wrist band (see Figure 2-2), such as a time line chart to show how mindful is the learners, a pie chart to present learner's concentration level during last two hours, a line chart to show the quantity of completed tasks. Feeler allows learners to self-regulate (change a task or take a rest) according to their physiological signals and help them to identify their own learning patterns.

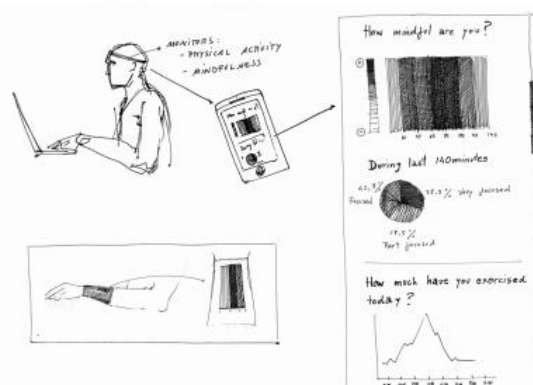


Figure 2-2 The dashboard of Feeler (Durall and Toikkanen 2013)

NAVI Badgeboard (Santos *et al.* 2013) aims at improving individual awareness and reflection of personal activities through visualizations of learners' communication activities (for example, the number of tweets and posts, the comments received by a blog) by "badge" presentations. Automatic trackers connect to the RSS feeds of the blogs and the Twitter API to collect learners' data and then store them in a database. It compares the quantities of learner's activities with the course goals by exploring the activity traces and presents these traces through badges when learners achieve one of the course goals. Badge visualizations help learners to be aware of their activities and compare their performances with the others' in the group. Personal Badge Dashboards (see Figure 2-3 a) contain a list of badges. The gray badges means the learner has not acquired yet while colored badges are opposite. A Class Progression View (see Figure 2-3 b) is supplied to learners to visualize the class's badge progression over time, which gives the learner a direct impression of how early or late he is at obtaining a specific badge compared to the others in the group.

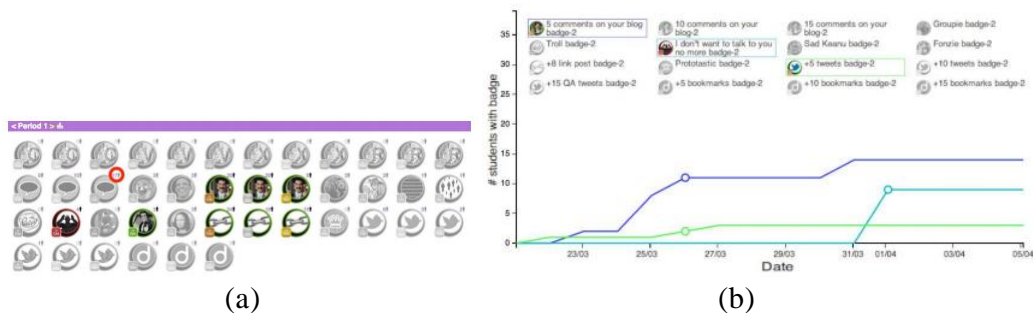


Figure 2-3 The dashboard of NAVI (Santos *et al.* 2013)

Phielix, Prins, and Kirschner (2010) proposed two tools, Radar and Reflector (see Figure 2-4). Radar is a peer feedback tool which supplies users with anonymous information about how learners themselves, their peers and the group judge of their cognitive and social behaviors. This tool calculates learners' performance from six aspects that are important for assessment in groups: influence, friendliness, cooperation, reliability, productivity and contribution quality. The first four are related to social or interpersonal behaviors while the latter two are related to cognitive behaviors. The learners can judge themselves and all the other group members based on the six principles by using a scale ranging from 0 to 4 (0 means none and 4 means very high) and each range is composed of 10 points. The results coming from group members are visualized in a radar diagram. There are two types of radar charts. One shows the results of the self-assessment and the average scores coming from the peer assessments. The other is about the average values of the group members, so that each group learners can know his position in the group and get a general view of the group.

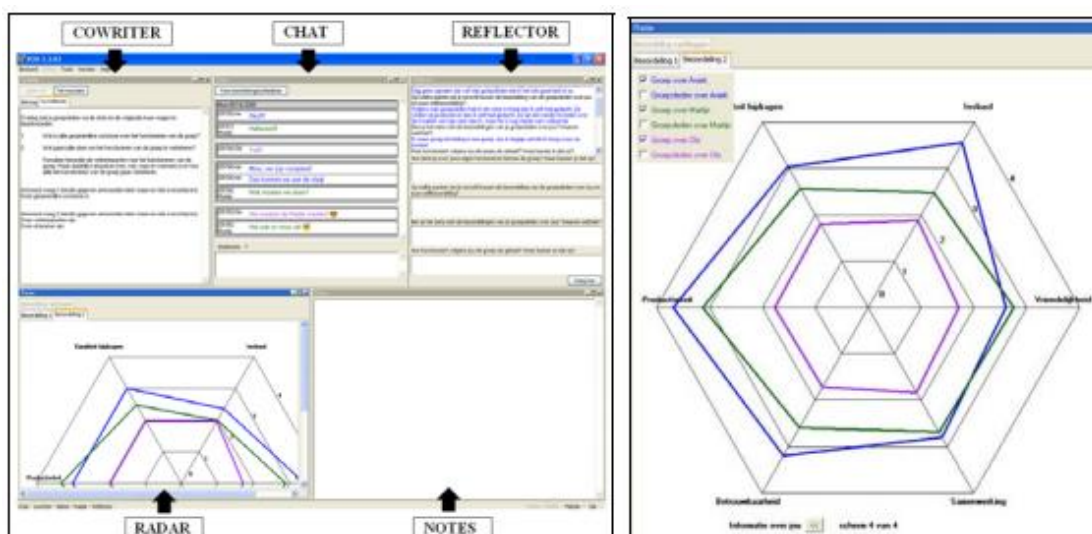


Figure 2-4 The dashboard of Radar and Reflector (Phielix *et al.* 2010)

Reflector (Phielix *et al.* 2010) is a reflection tool, which can encourage learners to reflect their individual behaviors by answering six reflective questions: “(1) *What is your opinion on how you functioned in the group? Give arguments to support this.* (2) *What differences do you see between the assessment received from your peers and your self-assessment?* (3) *Why do you or do you not agree with your peers concerning your assessment?* (4) *What is your opinion on how the group is functioning? Give arguments to support this.* (5) *What does the group think about its functioning in general? Discuss and formulate a conclusion that is shared by all the group members.* (6) *Set specific goals (who, what, and when) in order to improve group performance*” (Phielix *et al.* 2010).

LARAE (Charleer *et al.* 2014) provides the overview, context and content of the learners' traces to help them with awareness of feedback and progress, and assist tutors with monitoring learners' learning contributions so as to intervene when needed. Each group is composed of three learners and they can report weekly by blog posts, comments and twitters. The activities of posting, commenting, twittering and re-twittering are captured by system automatically. In the dashboard (see Figure 2-5), each type of activity is visualized by a circle and is sorted chronologically according to activity type and the learner group, which enable learners to be aware of the distributions of an activity thread across the class. The contents of the related activities (the content of the posts, comments and twitters) are display in thread views, which help tutors and learners to assess the quality of the quantitative data.

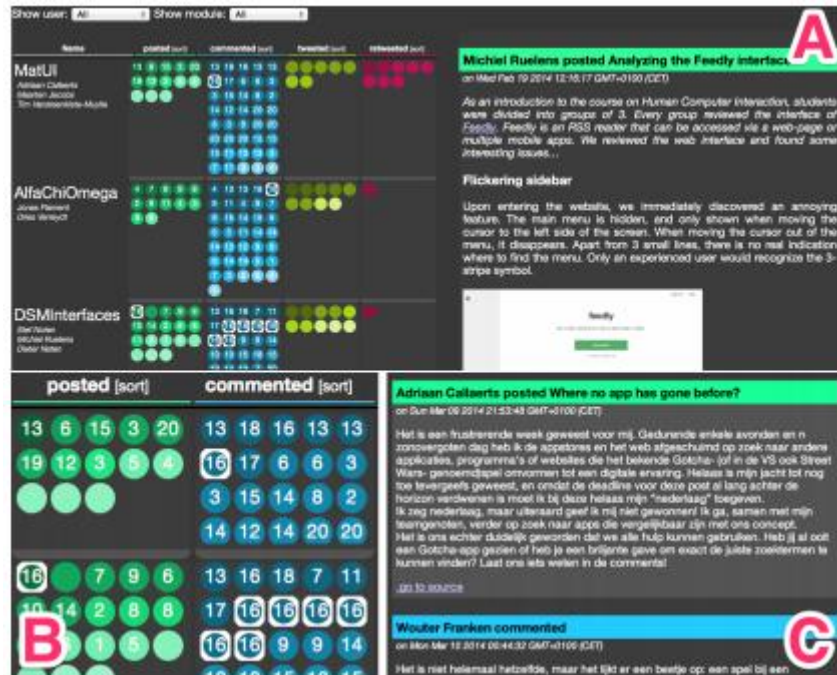


Figure 2-5 The dashboard of LARAe (Charleer *et al.* 2014)

Mastery Grids (Loboda *et al.* 2014) is a learning platform in computer science, which integrates open learning model (learner knowledge visualization) and social visualization (social comparison) in a form of social progress visualization to help learners to know how they are doing and where to allocate their efforts next. The dashboard supplies the social visualization based on three-dimensional grid. The horizontal dimension presents all the concepts in a domain of computer science (e.g., “if-else” or “Loops For”) while the vertical dimension presents all the learning resources supplied to learners (e.g., questions, examples, lecture notes, etc.). The third dimension is presented by the color intensity that means the level of mastery and the level of progress of a learner in a given concept and resource combination. Learners have two points of view in the dashboard. *All Resources mode* (see Figure 2-6) is integrated with (1) “Me” grid shows progresses the learner has made, (2) “Group” grid shows progresses that the currently selected group has made, and (3) “Me vs group” grid shows the differences between the two grids. *Resource Focus mode* focuses on one particular resource and presents the progresses the learner has made in this resource, the progresses the group has made in this resource as well as the differences between two. This mode makes the comparisons between me and group much easier and more directly. Each concept is composed of several activities. If a cell of the grid is clicked, activity grid is shown. Learners can know their own progresses, the group progresses and the differences on a specific activity.

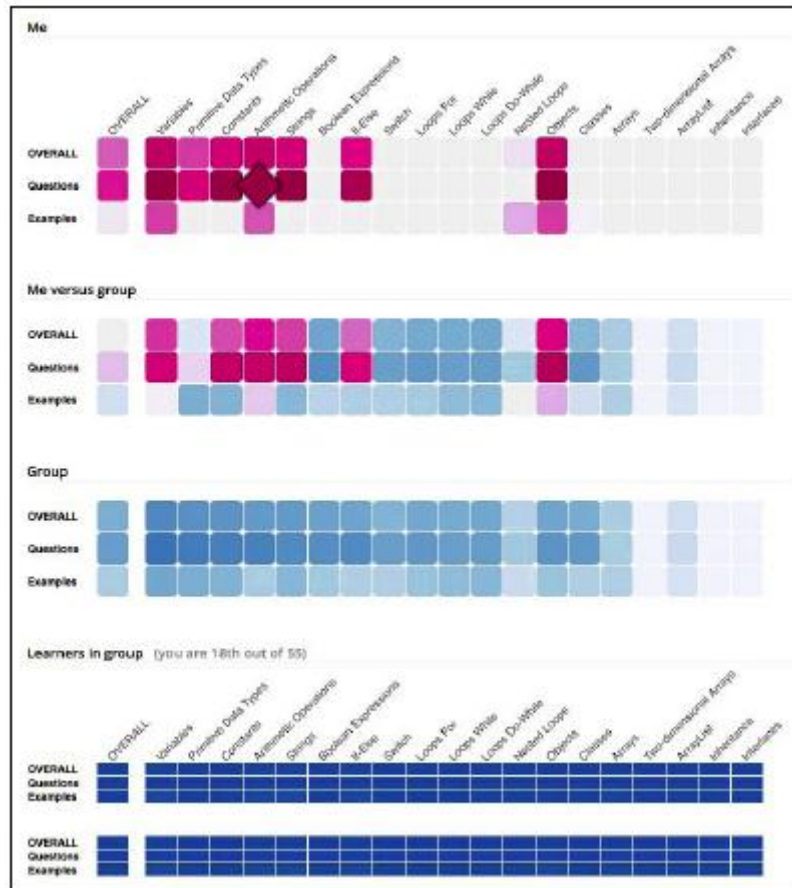


Figure 2-6 The dashboard of the Mastery Grids (Loboda *et al.* 2014)

Moodog (Zhang, Almeroth, and Knight 2007) is integrated into Moodle to track learners' online learning activities by exploiting available logs in Moodle. It can assist tutors to observe how learners interact with online project resources, as well as allow learners to compare their own progresses with the other members. It supplies statistics from four aspects:

1. **Course summary.** It presents the general information of a particular project, such as: the quantity of participated students, the amount of the available resources, the login time of each learner, the most (un)active learner and the most popular resource.
2. **Per-student statistics** (Figure 2-7 a). It allows learners/tutors to take a closer insight of a particular learner's progresses and compare between multiple learners, such as the times each resources are viewed by each learner, the amount of sessions each learner has, the time each student spend on Moodle, the quantities of the resources each student has or has not viewed and the total number of posts and comments wrote by each learner.
3. **Per-resource statistics** (see Figure 2-7 b). It allows the learners and tutors to know which resources have been viewed or not, such as how many learners have accessed a particular resource or the total view quantity of each resource.

4. **Time-based statistics.** It helps to identify the time issue (week, day and hour).



Figure 2-7 The dashboard of Moodog (Zhang *et al.* 2007)

GISMO (Mazza and Milani 2004) uses the log data of the learners' activities stored in Moodle, prepares the data for processing, and generates graphical visualizations that can be explored by tutors to become aware of social, cognitive, and behavioral aspects of learners. It explores the data from three aspects:

1. **Recourse view** (see Figure 2-8 a): shows the popularity of every resource, such as, the total access number of all resources viewed by each learner and the total access number of a special resource viewed by each learner every day.
2. **Discussion view** (see Figure 2-8 b): shows the discussions of learners. It supplies the information of every learner about the number of messages posted, the number of messages read, and the number of new threads posted by the learner.
3. **Login view** (see Figure 2-8 c): shows the accesses of the learners to the course. It supplies an overview of all login to Moodle of everyday by each learner and by whole group. Tutors can have a global view of the accesses to Moodle.

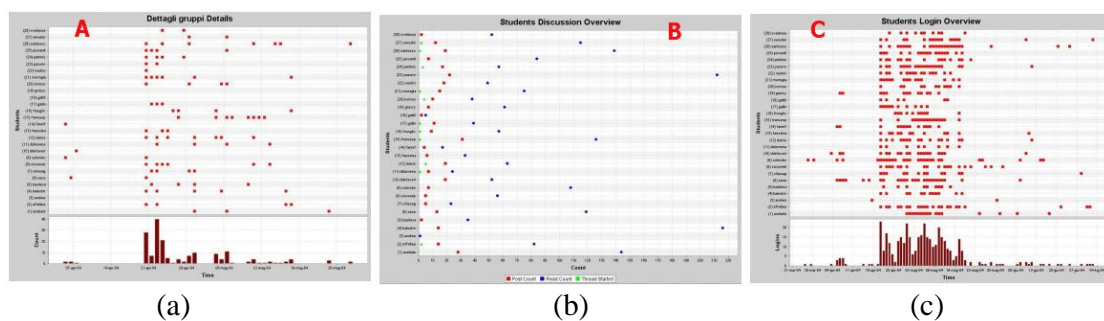


Figure 2-8 The dashboard of GISMO (Mazza and Milani 2004)

Govaerts (Govaerts *et al.* 2012) applied visualization techniques and proposed an application, named “Student Activity Meter” (SAM), which can visualizes learners' actions from the data tracked in learning environments. The overall goal of SAM is to assist tutors and learners with self-reflection and awareness of what and how learners are doing. SAM supplies five indicators.

The line chart (see label 4 in Figure 2-9) shows the cumulative amount of time spent by each learner. Every line represents one learner. A steep line means an intensive working period while a flat line shows inactivity. The users can compare between learners and find out how they spent their time on the activities. The users can zoom, filter and search a line. Parallel coordinates (see label 5 in Figure 2-9) are used to visualize different dimensional metrics together. Each learner is shown as a polyline. The vertical axes represent the metrics: the total time spent on the course, the average time spent on a resource, the number of resources used and the average daily frequency of doing an activity. The bar chart (see label 6 in Figure 2-9) shows the learner distribution based on the total time spent and the resources used. Box 2 in Figure 2-9 shows the statistics of time spent and documents use. It presents the statistics of a particular learner with the minimum, maximum and average time spent and documents used of all the learners. The recommendation pane (see label 3 in Figure 2-9) contains an animated tag cloud of the resources recommended to be used.

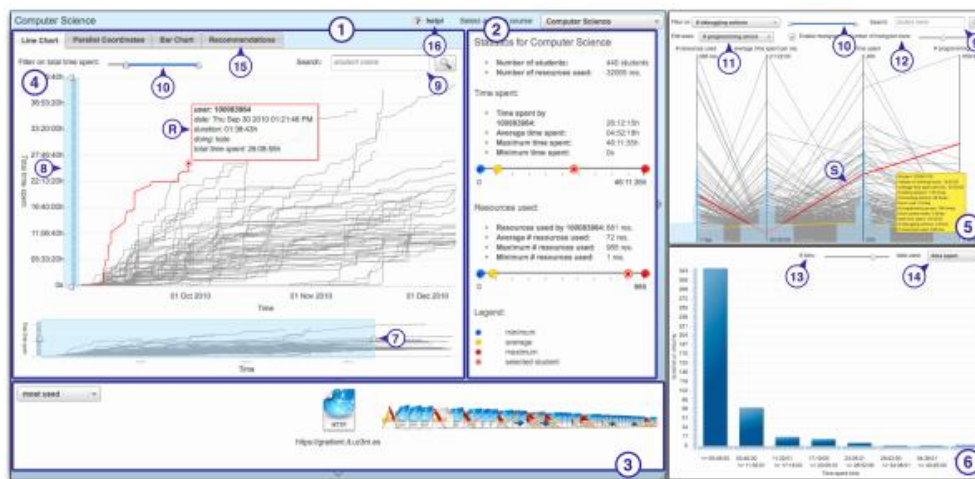


Figure 2-9 The dashboard of Student Activity Meter (Govaerts *et al.* 2012)

The observation of these nine simple dashboards leads to three conclusions:

1. **The reporting traces are ignored by most simple dashboards.** Most dashboards collect activity traces (e.g. Study Desk, Navi badgeboard, Mastery Grids, Feeler, Moodog, GISMO and SAM) to support self-regulation during learning activities while a small part of dashboards use reporting traces (e.g. LARAE, Radar and Reflector).
2. **The target user of the dashboards is learner.** Learners passively accept the information carried by the dashboards. Considering learners' abilities of acceptances, the balance between functionality and simplicity is nice. The indicators are predefined and no complex operation is required. Hence, learners do not need to have computer background.
3. **The customization level is low or even none.** Only LARAE and SAM supply the functions for learners to filter the indicator results. All of the other

dashboards cannot be edited by learners, even the visualizations of the indicators.

2.4.2 Dynamic dashboards with customizable indicators

Navi Surface (Santos *et al.* 2013) (see Figure 2-10) uses the same principle of badges presented with NAVI Badgeboard (mentioned in last section). Badge visualization supports group awareness, improves collaborations and leads to a better group reflection and individual reflection on learners' own activities in the group by exploring learners' activities traces. The customization level is low because learners can just set some simple filter parameters. Indeed, Navi Surface lists all the group learners and the badges available in the course, which can be touched and dragged. Learners or tutors can specify the names of the group members and select some badges to be presented into the "Playfield" in order to view how the group badges are acquired by presenting the relationships between the badges and learners. This tool has user-friendly interfaces and so can be used easily by all the types of users. However, the only visualization mode is badge.

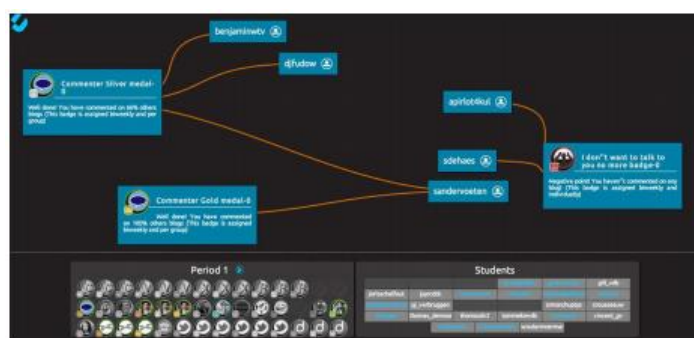
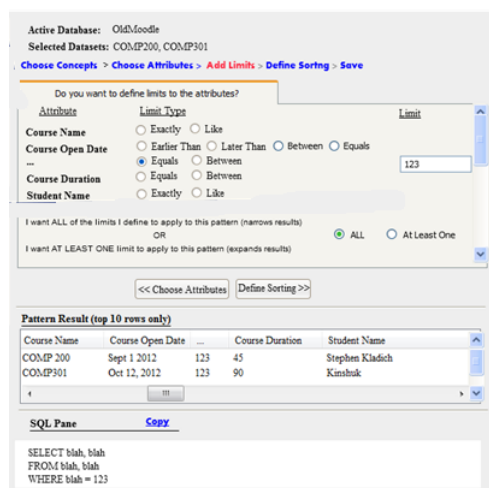


Figure 2-10 The dashboard of Navi Surface (Santos *et al.* 2013)

The Academic Analytics Tool (AAT) (Graf *et al.* 2011) (see Figure 2-11) allows tutors to analyze learners' behavior data. It explores the traces about how learners interact with each other and learn from online courses in a learning system, which enables tutors to analyze the extracted traces, and to store the results in a database. The traces used for calculation are log traces recorded by the system. AAT supplies tutors with medium customization functions by allowing them to extract specific information from the activities 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 etc.) but also create custom ones. However, the only visualization mode of the results is table. Tutors are required to write SQL queries. So specific computer skills are needed.

Figure 2-11 The dashboard of AAT (Graf *et al.* 2011)

GINDIC (Gendron 2010) (see Figure 2-12) can help tutors to create and manage customizable indicators. Tutors need to define the indicators and implement the calculation on the traces in order to get the indicator value and the visualization. It divides this process into four steps: (1) defining the concept and basic information of the indicator (name, description, point of view etc.); (2) constructing all the parameters for building an indicator by selecting the data sets, selecting the rules, adding filter, editing calculations and so on; (3) contextualizing the indicator by declaring the target users' rights (delete, modify, visualize) according to their roles; (4) selecting one or more visualization modes for the indicator. These four steps support the generation of medium customizable indicators by allowing users to create calculation methods and assign different visualization modes to a particular indicator, but require the users to have a computer background because many parameters need to be set (for example, select rules, editing calculation operators, assign the rights etc.).



Figure 2-12 The dashboard of GINDIC (Gendron 2010)

The TBS-IM system (Djouad and Mille 2010) (see Figure 2-13) uses the concept of modeled trace to enable learners to create indicators from the activity

traces produced by Moodle about learners' activities and learning progresses. The modeled traces are organized according to a trace model. The trace model defines and structures explicitly the sequentiality and content of the trace (Settouti and Prie 2009). A single, common and extensible trace model can be used for several modeled traces and help to reduce the number of tools needed to explore them. Some transformation operators are supplied to users (fusion, rewrite, select, delete etc.), which enable users to transform low level (fine granular) traces into high level (synthetical) traces. The customization process is organized into three steps: selection, transformation and visualization. The customization level is medium. Tutors can pick up the calculation operators to extract more comprehensive information, and so forth. At last, TBS-IM offers some visualization modes for users (e.g. bar chart, line chart, pie chart) to select.

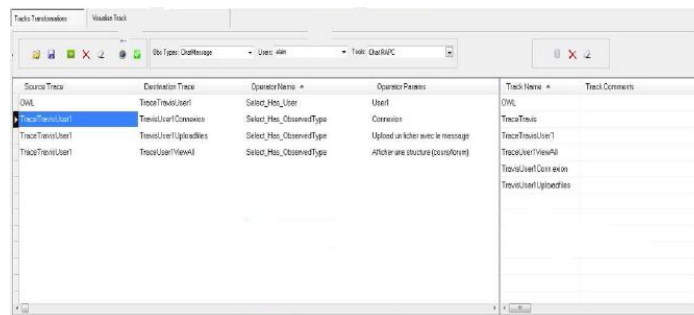


Figure 2-13 The dashboard of TBS-IM (Djouad and Mille 2010)

2.4.3 Discussion

According to analysis above, in Table 2-4, we compare the previously presented dashboards from four aspects: trace type, target user, customization level and the requirements of computer background.

Table 2-4 Comparison of the dashboards

Dashboard	Trace type	Target user	Customization level	IT requirements
With predefined indicators				
Study Desk	Activity	Learners	No	No
Radar and Reflector	Reporting	Learners	No	No
LARAE	Activity /Reporting	Tutor /Learner	Low	No
Feeler	Activity	Learner	No	No
NAVI badgeboard	Activity	Learner	No	No
Mastery Grids	Activity	Learner	No	No
Moodog	Activity	Tutor /Learner	No	No

GISMO	Activity	Tutor	No	No
SAM	Activity	Tutor/Learner	Low	No
With customizable indicators				
GINDIC	Activity	Researcher	Medium	Yes
AAT	Activity	Tutor/ Researcher	Medium	Yes
TBSIM	Activity	Learner	Medium	No
Navi Surface	Activity	Tutor /Learner	Low	No

From the above table, some conclusions can be drawn:

1. ***Most dynamic dashboards are dedicated to tutors and researchers while most simple dashboards are supplied to learners.*** The learners are considered as information accepters rather than information creators or information explorers. So it is difficult for them to self-regulate their activities and to build metacognitive skills because they are not allowed to access the processes of creating indicators. Hence, they are not supported to monitor their performances in their own ways.
2. ***Most dashboards explore the activity traces to analyze the learning processes while the reporting traces are ignored.*** The activity traces are well structured and can be captured and analyzed automatically. However, they cannot represent the learning quality or the learning processes completely. For example, the reporting traces are needed to analyze learners' states of minds, the quality of the productions, the processes of carrying out a collaborative activity out of the learning system, the assessments between the group members, etc. It is proved that students perceive higher usefulness if dashboards present a more complete description of their learning activities (Verbert *et al.* 2013). The whole learning processes can be observed globally if the indicators are created based on two types of traces.
3. ***Most simple dashboards do not supply the customization functions or just supply some low customization functions while the dynamic dashboards have medium customizable functions.*** However, for dynamic dashboards, many parameters are required to be set and users cannot add new information into the traces. Most simple dashboards supply a single visualization mode to each indicator while some dynamic dashboards supply various possible visualizations for an indicator. The variety of visualization modifies the capacity of the information presented into indicators to be understood.
4. ***The higher the customization level is, the higher the computer background requirement is.*** There is no simple, user-friendly and high-dynamic dashboard proposed to learners. It is a critical problem that needs to be considered carefully.

2.5 Conclusion

In this chapter, we defined the context of PBL and SRL. Throughout this study, we show that PBL is an effective and learner-centered method to help learners to build new knowledge and get new skills. This kind of activity can lead learners to obtain and apply skills in collaboration, communication and self-management. However, many researches proved PBL is difficult because learners are lack of self-regulation skills. That is to say, enabling learners to self-regulate their learning and project is the key to success in PBL. We found that most E-learning systems could be used in PBL by supplying with plenty of tools that could help learners to plan, monitor and assess of the projects and their performances. Among the methods to support SRL, we think that it is a direct and effective way to build indicators by analyzing the traces left by learners during the projects. Further, we analyzed two different traces: activity traces and reporting traces. It has been proved that each type of trace has its own strengths and limitations. We hypothesize that the combination of activity traces and reporting traces can support to create more helpful indicators and enable learners to explore their projects and learning more effectively. It has been affirmed that the dashboard is effective to present indicators from an explicit and synthetical point of view. We compare several simple dashboards with dynamic dashboards. We observe that most dynamic dashboards are targeted on tutors or researchers and the ones, witches are directed toward learners, have a low customization level or require high computer backgrounds.

In the next chapter, we propose a Project-based Learning Management System (PBLMS) framework that can improve learners' self-regulation and meta-cognition skills during PBL by supporting them in the creation of customizable indicators related to their activities. This framework collects both activity traces and reporting traces. Learners can build their own customizable dashboards by exploring these traces. It is a four steps framework: data collection, data integration, indicator calculation and indicator visualization.

3 A Project-based Learning Management System

In this chapter, we propose a general framework of Project-based Learning Management System (PBLMS). The aim of a PBLMS is to help learners to carry out a project by developing their self-regulation skills. We define the functions and the modules of this kind of system.

3.1 Proposition of a general framework of PBLMS

A Project-based Learning Management System (PBLMS) can be defined 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*” (Ji 2012). We detail hereafter the different components a PBLMS has to equip (see Figure 3-1).

1. **Collaboration.** A project is collaborative and learners cannot accomplish a project alone. Hence, a PBLMS has to support learners to work collaboratively. According to the different aspects of projects, different kinds of collaboration tools are needed, for example, plan tools (work divisions, project schedule, project goals, etc.), communication tools (chat, video conference, etc.), knowledge sharing tools (wiki, glossary, forum, etc.), writing tools (group reports, group blogs etc.), special and professional tools (modelling, robotics, etc.). These kinds of tools are generally provided by the existing Learning Management Systems.
2. **Reflection.** During the project, learners have to reflect on the project processes. Reflection can help learners to structure the way they carry out their projects and encourage them to gain internal feedbacks. Reflection tools have to supply learners with means to record the activities that the collaboration tools do not support, for example self-assessment and peer-judgement.
3. **Self-monitoring.** Self-monitoring tools help learners to monitor their own behaviours, performances and judgments. Learners can extract information about their collaboration and reflection activities and visualize this information to improve their projects and learning. The self-monitoring tools have to be customizable to allow learners to choose the indicators that are the more useful for them.

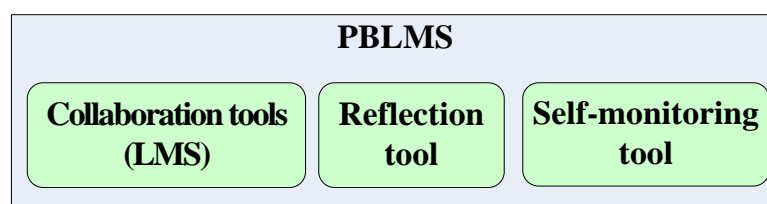


Figure 3-1 The components of PBLMS

Most of the existing learning systems focus on how to support learners to carry out the projects collaboratively but do not support self-regulation processes. Compared with the functions supplied by classical learning systems, PBLMS can not only help learners to carry out the project activities collaboratively but also can facilitate them to gain reflection and self-monitoring skills.

We distinguish two types of project activities that can be observed to generate different kinds of traces:

1. ***The automatic traceable activities.*** When learners use the tools supplied by the PBLMS to accomplish some collaborative activities, such as chatting with others, sending private messages, joining in a videoconference, uploading/ downloading resources, the requests sent to the server side of PBLMS can be captured. Hence, these activities can be tracked automatically by the systems.
2. ***The non-automatic traceable activities.*** According to the context of the project, learners could need to use special tools that are not offered by PBLMS, such as MS Office, Firefox, Eclipse, MySQL etc. Furthermore, some projects could require learners to do some outdoors activities (e.g. interviewing in the street, collecting leaf examples in the forest, observing the production processes in a factory), non-instrumented activities (e.g. face to face discussions), reflection activities and assessment activities. All of these activities cannot be tracked automatically. Hence, in order to present the information related to the project and to help learners to reflect, they have to record these non-automatic traceable activities manually.

The automatic traceable activities can be captured directly by the PBLMS as traces when the learners send the service requests to the system server side (e.g. enter into a chat room, click a button to submit a form). We call this kind of traces ***activity traces***. The non-automatic traceable activities are unpredictable and diverse because projects require learners to accomplish different activities. Hence, we propose to integrate a reporting tool to support learners' reflection processes. They so can record all of the non-automatic traceable activities manually. We define this kind of traces as ***reporting traces***.

Djouad and Mille (2010) proposed three processes to manipulate the traces to create indicators: collecting data (selecting related and useful data within the tracing sources), data transformation (performing operations on traces like using filters, rewriting, fusion, to generate transformed traces) and indicator calculation (calculating the transformed traces). However, the reporting traces are not taken into account in the whole processes. Furthermore, the calculation operators enable to do only basic calculations (e.g. count, addition, subtraction). It is necessary to supply more complex calculation methods in order to produce more synthetic and complex indicators. The indicator visualizations are also limited. In Figure 3-2, we propose a global architecture of PBLMS composed of

four modules, which exploit both reporting and activity traces, allow learners to add data into traces, do complex calculations, and personalize the indicator visualizations.

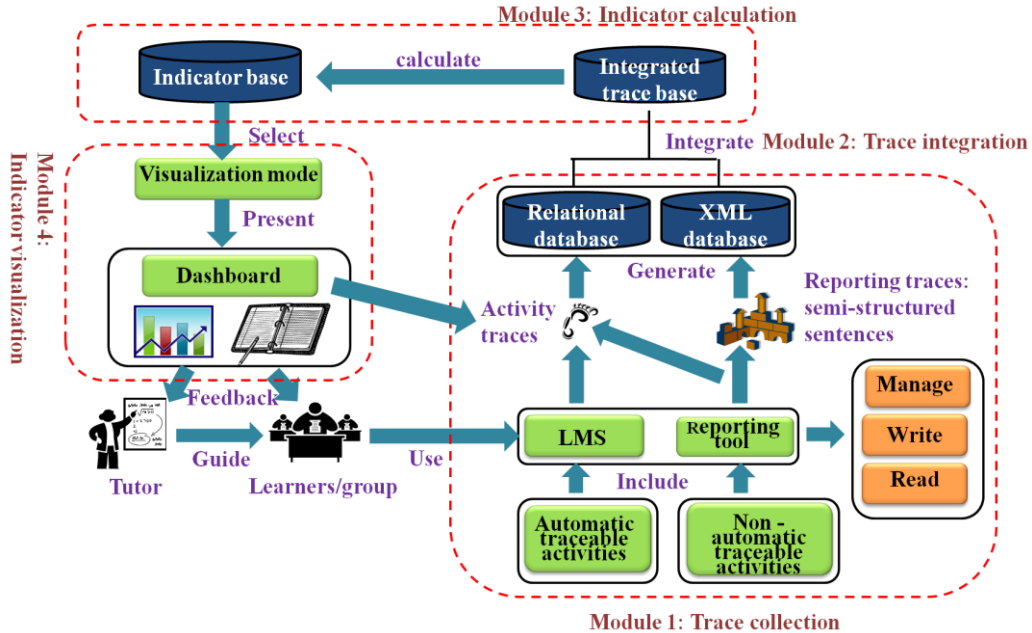


Figure 3-2 The framework of PBLMS

Module 1. Trace collection. This module captures or collects the activity traces and reporting traces from different sources. For the activity traces, the automatic traceable activities can be tracked directly by the PBLMS and be stored in a relational database according to the activity trace models (see section 3.3.1). The reporting tool enables learners to write reports to record their non-automatic traceable activities by using semi-structured sentences (see section 3.3.2).

Module 2. Trace integration. This module integrates activity traces and reporting traces together based on a uniform integrated trace model (see section 3.4). The traces are stored in an integrated traces base, which are the data sources to generate indicators.

Module 3. Indicator calculation. This module produces indicators based on the integrated traces by applying different calculation methods (see section 3.5). The calculation methods can aggregate the integrated traces from low (fine) level into high (synthetic) level. In order to create a customizable indicator, some parameters are required to be set, such as the calculation methods, the information to be observed.

Module 4. Indicator visualization. A PBLMS with customizable indicators should supply learners with multiple visualization modes (see section 3.6), which allows learners to observe the data from different dimensions. For example, line chart presents a trend according to timeline, bar chart enables to

compare between different data sets. All the indicators are presented within an individual dashboard.

We sum up the characteristics of PBLMS as follows:

1. ***It integrates Learning Management System (LMS).*** LMS provides a variety of collaborative tools (such as communication, knowledge sharing, document download/upload etc.) to support a range of activities and is widely used in universities. We decide to take advantage of these tools in PBLMS to support collaboration.
2. ***It enables learners to record information about their activities and reflections.*** By using a reflection tool, learners can record the information that cannot be tracked automatically by PBLMS, such as: the non-instrumented activities, their reflections on the activities or their assessments. This information can be generated into reporting traces to be analyzed.
3. ***It integrates activity traces and reporting traces.*** The activity traces and reporting traces can be integrated together to produce indicators. These two kinds of traces complement each other and present a global view about the project processes. A uniform data model is necessary to integrate these two types of traces and enables to store the traces.
4. ***It supplies a dynamic dashboard.*** A dashboard can gather all the indicators together to supply learners with a general view at a glance. A dynamic dashboard allows learners to create customizable indicators (data, calculation and visualization) by exploring the activity traces and reporting traces.

The objective of the proposed framework is to support learners in the creation and the management of customizable indicators by exploring the traces of their activities and reflections. In the next part, we focus on the indicators management according to the proposed framework of PBLMS.

3.2 Indicator management

We define an indicator (I) as:

$$I = \{P, V, IT\}$$

With:

1. P: the parameters of the indicator, including the information to observe (see section 3.3 and section 3.4), the data types and the calculation operators (see section 3.5).
2. V: the indicator visualization diagrams (see section 3.6).
3. IT: the integrated traces (see section 3.4) used to produce the indicator.

This definition means that an indicator is a specific diagram chosen to present the calculation results of the integrated traces computed with specific parameters.

A PBLMS enables learners to manage indicators from three aspects: creating an indicator, updating an indicator and deleting an indicator.

1. **Creation of an indicator.** Learners can create customizable indicators. This process is divided into four steps (see Figure 3-3): selection of the information to observe from the integrated traces, selection of the calculation methods to apply on the integrated traces, selection of one or several visualization mode(s) to present the calculation results into diagrams, and saving the indicator in the database.

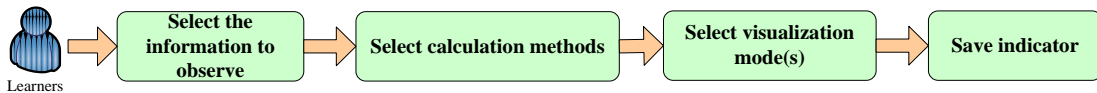


Figure 3-3 The process of creating a new indicator

2. **Updating an indicator.** Learners can update their own indicators or reuse the public indicators created by others (see Figure 3-4). Learners have to select an indicator they want to update. They can modify the observed information, the calculation methods and/or the visualization modes. If the learners are not the indicator creators, they have no rights to modify the original indicator and they only can save this modified indicator as their own new indicator. If the learners are the indicator creators, they can save it as a new indicator or update it.

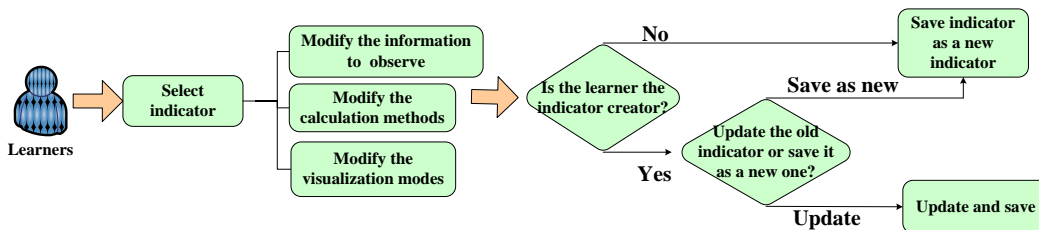


Figure 3-4 The process of updating an indicator

3. **Deletion of an indicator.** Learners can delete the indicators they have created.

In order to guide learners in the creation of different kinds of indicators in PBLMS, we propose a classification of indicators. Our classification relies on several works. For instance, Dimitrakopoulou *et al.* (2006) classified the indicators into three categories according to the indicator purposes: cognitive, social and affective. Cognitive indicators refer to cognitive operations related to the learning activity content and the activity process. Social indicators are related to the communication, cooperation, or collaboration activities in a group and the affective indicators reflect the emotional situations of the learners. Diagne (2009) proposed three different categories: cognitive, activity and social. The cognitive indicators provide information on the knowledge acquisition in learning activities. The activity indicators provide information on the usage of educational resources and computer tools in the learning platform. The social

indicators focus on the relationships between group members, learners' positions in the group and their participations in collaborative activities.

Comparing the two classifications, we find that Dimitrakopoulou *et al.* (2006) integrated the information of activity process into the cognitive category. However, it seems more appropriate to differentiate the product and the process of the activity and so to distinguish between cognitive and activity indicators. That enables learners to focus on one aspect of the activity. Furthermore, Diagne (2009) did not consider the affective category, which appears necessary to reflect learners' emotional states during the project. Accordingly, we classify the indicators, which have to be supported in PBLMS, into four categories: activity, cognitive, affective and social.

1. **Activity indicator.** It presents the information about activity processes and learners' behaviours: e.g. the time spent on the work, the density of activities in a period of time, the quantity of each tool used by the group members, the activity schedule compared with the activity in practice.
2. **Cognitive indicator.** It reflects the information about learners' process of acquiring knowledge/skills, which relates to the goals set by the learners at the beginning of the project (e.g. the level of a specific knowledge acquired by a learner compared with the planned level). It also reflects the information of the project results, e.g. the content of the project wiki (or glossary) edited by the group members.
3. **Affective indicator.** It represents the state of minds of the group members during the project: e.g. the emotion trend of a particular learner in a period of time. This type of indicators can monitor learners' affective changes.
4. **Social indicator.** It produces information about the interactions between group members and the impressions of each learner coming from peer-judgement. This information reflects the state of collaboration and social organization in a group: e.g. social network analysis and peer judgments.

In order to be able to create these four types of indicators, the PBLMS has to collect the traces related to the activity, cognition, affection and social process during the project. The trace collection module proposed in section 3.3 aims at collecting all these kind of traces. We more precisely propose several semi-structured sentences that enable learners to report different kinds of information.

In the following sections, we explain in details each module that composes the PBLMS framework: trace collection, trace integration, indicator calculation and indicator visualization.

3.3 Trace collection

The trace collection module collects the traces, including activity traces and reporting traces. Activity traces are left by learners when they interact with the

PBLMS. This part of traces can be tracked automatically by the PBLMS. Reporting traces are reported manually by learners in the reporting tool with the form of semi-structured sentence models. According to the different types of traces, we design different methods to structure and collect them.

3.3.1 Activity traces collection

The activity traces correspond to the interactions between learners and the PBLMS and the interactions among learners supported by the PBLMS. The Figure 3-5 presents the architecture of the collection module of activity traces, which can be divided into two parts: browser side and server side.

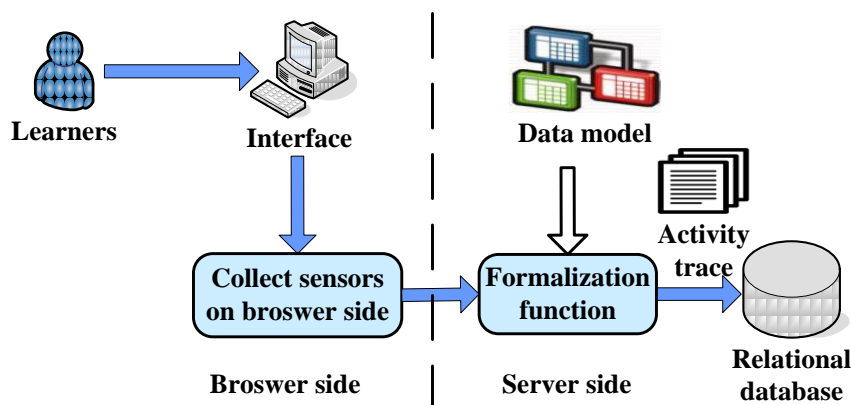


Figure 3-5 The architecture of activity trace collection process

At the browser side, learners interact with the PBLMS through the interfaces, for instance by writing in a forum, editing wiki pages, viewing a resource or sending a message. The collect sensors enable to capture these browser events (e.g. page load/close, textbox input, button click). The data of these events captured by the collect sensors are used to generate the activity traces. Considering that the LMS is imported into the PBLMS, some collect sensors are already developed in the LMS. If the data collected by the default collect sensors in the LMS are insufficient, it is necessary to develop new collect sensors.

At the server side, when the server receives the data from the collect sensors, the formalization functions are called. All the received data are formalized by a set of data models, which are corresponding to the different interactions, to generate activity traces. The data models define the structure of the activity traces, such as the classes, the relationship between classes, the attributes of the classes. The activity traces are stored in a relational database. We choose to use a relational database because most LMS use this kind of database. It will so be easier to integrate the activity traces already collected by the LMS with new activity traces.

Project can be conducted in different contexts. So it is difficult to specify which specific types of activity traces are required to be captured by the

PBLMS. We give an example of the activity “sending a chat message” to explain how the system captures the activity traces (see Figure 3-6). Learners write a chat message in a form presented in the PBLMS interface. After they click on the submit button, the corresponding collect sensor in the server side is triggered. The information related to the interaction is collected and is sent to the corresponding formalization function in the server side. The server side formalizes the received information according to the data model (see Figure 3-7) and connects to the database to store the interaction traces. If the chat message is stored successfully, a message returns back to the PBLMS interface to inform learners.

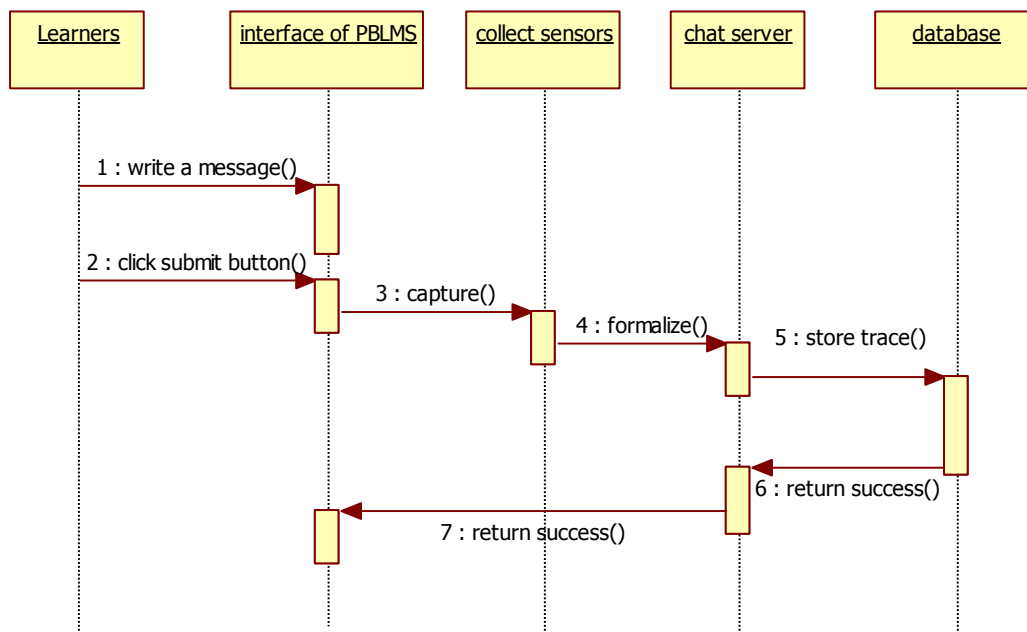


Figure 3-6 An example of the activity “sending a chat message”

Figure 3-7 is the entity-relationship schema of the activity trace of the activity “sending a chat message”. There are three entities: the user entity is used to store the basic information on the learners, the chat entity records the basic information on the chat tool, and the message entity stores the contents of the chat messages. Table 3-1 presents an example of the activity trace of this activity.

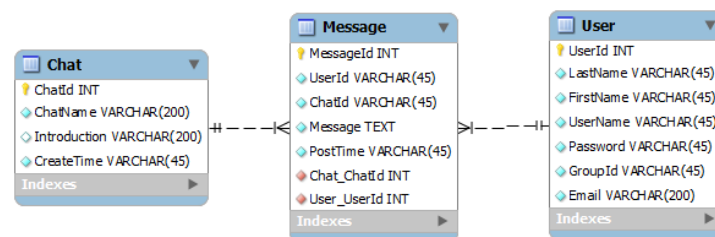


Figure 3-7 The entity-relationship schema of “sending a chat message”

Table 3-1 An example of the activity trace “sending a chat message”

Table User						
UserId	LastName	FirstName	UserName	Password	GroupId	Email
001	Amy	Smith	amys	*****	1	amys@gmail.com
Table Chat						
ChatId	ChatName	Introduction			CreateTime	
01	Chat room 1	This chat room is used during the project planification.			01-01-2015	
Table Message						
MessageId	UserId	ChatId	Message		PostTime	
001	001	01	“Hello!”		02-01-2015 09:12:23	

3.3.2 Reporting traces collection

We propose a reporting tool to collect the information related to the non-automatic traceable activities, such as the activity processes, the learners' states of minds, the learners' assessments, the learning strategies and the learners' objectives. All of this information cannot be tracked by the PBLMS automatically. Hence, we propose the concept of semi-structured sentence models.

Figure 3-8 presents the architecture of the reporting tool. Learners can select several semi-structured sentence models they want to fill in. The selected semi-structured sentences completed by learners compose a report. The reports are stored as XML files in an XML database. We chose an XML database to store the reporting traces according to the following reasons:

1. The reporting traces are composed of text and the structures of the reports are different. Compared with a relational database, XML database can better support the text data that is less structured.
2. XML has semantic tags to represent the meaning of the data. So it is convenient to process and “understand” these data. For example, in the XML element “<messageContent> How are you? </messageContent>”, the tag “<messageContent>” indicates that the element content (“How are you?”) is the content of a message.

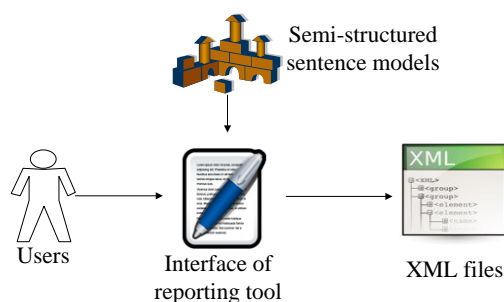


Figure 3-8 The architecture of the reporting tool

A semi-structured sentence model is composed of two parts. One part is the fixed part that organizes the structure of the sentence. The other parts are variables, and learners can assign values by choosing in a predefined list or filling in the text field. For example, a semi-structured sentence model can be presented as “*I do what today at where*”. Learners can fill in this semi-structured sentence model by specifying the content of the activity and the place of the activity.

We classify two types of reports provided by the reporting tool: the goal reports and the activity reports.

The goal report is written at the beginning of the project to help learners to plan their projects and to set the goals they want to achieve at the end of the projects. It supports the forethought phase of SRL. We propose three semi-structured sentence models that can help learners to focus on three aspects: the project goals, the skill and knowledge goals and the project schedule.

1. **Project goals.** Learners report the goals of the project by describing the project objectives, the duration and the content. It helps learners to organize and reflect on the project goals, to compare their own understandings of the project goals with the other group members and to adjust their views if necessary.
2. **Skill and knowledge goals.** Learners can set the levels of skills and knowledge that they want to acquire during the project. It helps them to compare their skills (knowledge) goals with their current levels of self-assessed skills (knowledge). They can know whether they have reached their goals or not.
3. **Project schedule.** It enables learners to schedule the project activities by declaring their contents and the time scales. The indicators can be built by comparing the planned activities schedule with the real activity schedules. Learners can so observe how they carry out the project and whether they follow their schedules or not.

The activity reports are used to record learners' non-automatic traceable activities and support reflection processes of SRL. By completing the semi-structured sentences, learners can describe the project activities carried out without using the PBLMS, self assess, and judge the other members. The semi-structured sentence models are as follows:

1. **Activities.** It is used to report the project activities that cannot be captured automatically by the PBLMS, such as outdoors activities and activities carried out with other tools (e.g. MS Word, Skype, Gmail). This model covers a range of activity information, for example activity contents, time, place, actors and collaborators. This kind of information can be used to build both activity indicators (on the project implementation), and social indicators (on the interactions in the group).
2. **Self-assess.** Learners can assess their levels of acquisition of knowledge and/or a skill at a given step of the project. This information can be used to build cognitive indicators by helping learners to compare these levels with their goals. Learners can also assess their states of minds during the project. The affective indicators can be built based on this type of information.
3. **Judgment.** Learners can judge the other group members or an object related to the project, such as a book, a web site, a tutorial video or an academic paper. It enables to build social indicators based on learners' peer judgments.

We propose a class model of the reporting traces (see Figure 3-9). It presents the data structure, which can support the data storage and structuration. It is composed of seven classes: Report Content, Semi-structured Sentence, Report Structure, Variable Type, Variable Value, Category and Customized Structure.

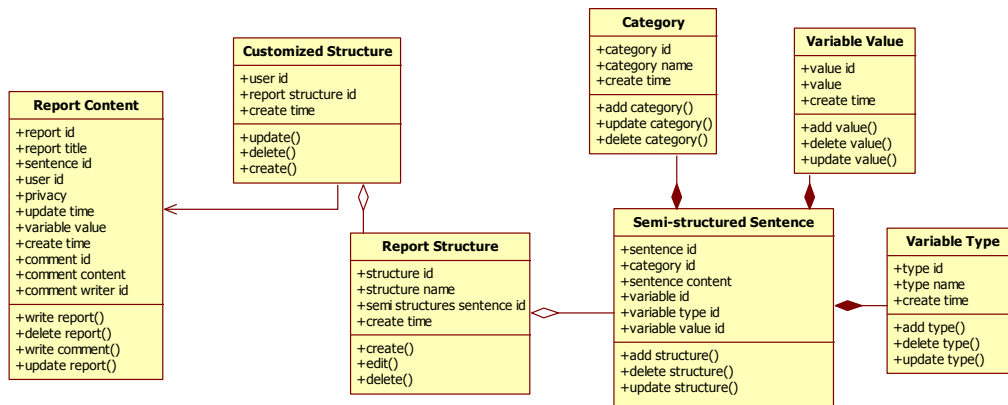


Figure 3-9 The class model of the reporting traces

In Table 3-2, we explain each class that composes the class model of the reporting traces.

Table 3-2 The class description

Class	Description
Report Content	The class contains: <ol style="list-style-type: none"> 1. The basic information of the reports, such as writer id, time, privacy and report title. 2. The contents of the reports, such as the semi-structured

	sentence id, the fill in words of the sentences. 3. The comments of each report. The comments given by other group members.
Semi-structured Sentence	It consists of the semi-structured sentence models and describes the structures of each sentence model, such as the optional values of variables, the type of each variable, the category of each model, and the fixed parts of models.
Category	It describes the category of each semi-structured sentence model, such as plan, activity and judgment.
Variable Type	This class describes the type of each variable in each semi-structured sentence model, such as text area, dropdown box, etc.
Variable Value	It describes the optional values of the variable. For example, when the variable type is dropdown box, all the possible values are stored in this class.
Report Structure	It maintains the relationship between the Customized Structure class and the Semi-structured Sentence class. A report structure consists of several semi-structured sentences models according to the different purposes of the report.
Customized Structure	It assigns different report structures to different learners. Learners have their own customized report structures according to the project time or to their profiles.

Activity traces and reporting traces are heterogeneous. In order to build indicators based on the activity traces and reporting traces, we have to integrate these two types of traces together. In the next section, we propose a uniform integrated trace model that integrates the two types of traces.

3.4 Trace integration

The integration module aims to aggregate the two types of traces and to generate the integrated trace (IT). The integrated traces are the source data for creating indicators and are structured by an integrated trace model. Figure 3-10 presents the process of trace integration. The activity traces stored in the relational database and the reporting traces in the XML files are processed according to an integrated trace model that generates the integrated traces.

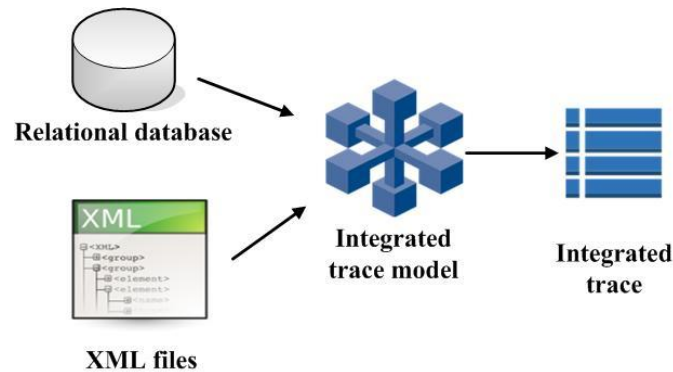


Figure 3-10 The process of trace integration

The integrated trace model can be described as follow:

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

With:

1. Id: the trace id.
2. A: the activities carried out by learners during the project and the subjects of the plan or judgment, e.g. view forum, chat with others, write report, take a course, organize a brainstorm and the skill assessed by the learner.
3. L: the learner who carries out the activity. It can be any member of the project group.
4. Ca: the category of the activities. The activities are classified into three categories: planification, project tasks and judgment.
5. Co: the content of the activities. It includes the detailed content of the activity (e.g. the content of forum posts or chat messages sent by a learner, the content of the wiki written by a learner), the declaration of the judgment, the topic of a meeting, the content of the plan.
6. To: the tools used by learners to accomplish the activities, e.g. wiki, chat, forum, Dropbox, Firefox, Netbeans.
7. P: the place where learners accomplish the activity, e.g. home, university or a virtual place.
8. RO: the related object of the activity: the collaborators of the activity or the related trace id, for example, if a learner views a forum post, the trace id of writing this forum post will be listed here.
9. BT and ET: the begin time and the end time of the activities.
10. C: the comment of the traces, which presents the complete reporting information.

The attributes of this model enable to describe the different aspects of the project activities, reflections and assessment. That allows learners to explore the traces by focusing on different points of views. In Table 3-3, we give the examples of integrated traces.

Table 3-3 Examples of integrated traces

Trace Model	Integrated trace 1	Integrated trace 2	Integrated trace 3	Integrated trace 4
Id	T210	T211	T212	T213
A	View forum post	Read an academic paper	PHP	Julien
L	Tom	Tom	Tom	Tom
Ca	Project task	Project task	Planification	Judgment
Co	-	How to do project requirements analysis	High	Low
To	Forum	Adobe reader	Reporting tool	Reporting tool
P	Moodle	Home	Moodle	Moodle
RO	T132	Mary	-	-
BT	01-04-2014 08:00	01-04-2014 09:00	02-04-2014 09:00	01-05-2014 09:00
ET	01-04-2014 08:10	01-04-2014 09:30	02-04-2014 09:00	01-05-2014 09:00
C	-	I read an academic paper about how to do project requirements analysis with Mary by Adobe reader at home from 01/04/2014 09:00 to 01/04/2014 09:30.	I plan that my PHP should reach the high level.	My mood is low because I am tired.

We consider that the attributes “Ca” (category) and “Co” (content) depend on the attribute “A” (activities). The attribute “Id”, “RO” (related object), “C” (comment) has no practical meanings and the attributes BT (BeginTime) / ET (EndTime) can be considered together. We so extract five significant elements from the integrated trace model: A, L, To, P and Ti (BT and ET), which correspond to what, who, how, where and when. We rename these five elements as *entities*. Each entity is independent from the others and has practical meanings. The values of the first four entities (A, L, To, P) are extracted from the integrated traces. The values of “Time” entity are presented by three time scales: day, week and month. Learners can select the various time scales according to the granularities of the observation they want to do. When building an indicator, learners can choose different entity values according to their needs and the information they want to observe.

In this section, we proposed an integrated trace model to combine the activity traces and reporting traces together. We more precisely focused on five entities that learners can use to build indicators from different points of views. In the next part, we describe how to manipulate the integrated traces and calculate the indicators from fine level to synthetical level by using a set of calculation methods.

3.5 Indicator calculation

The indicator calculation module provides the selection of entity values, their data type and the calculation operators to apply on them. The different possible

combinations of entity values, data types and calculation operators offer many possibilities for learners to build indicators. We describe in this section the data selection process, the data types and the calculation operators.

3.5.1 Data selection

The data selection process consists in choosing the entity values that compose the indicator. There are two possible operations: crossing and filtering.

3.5.1.1 Crossing

The crossing operation consists in calculating the Cartesian product of two entity value sets. It can be defined as follows:

$$E_1 \times E_2 = \{E_{11}, \dots, E_{1n}\} \times \{E_{21}, \dots, E_{2m}\} = \{E_{11}E_{21}, \dots, E_{11}E_{2m}, E_{1n}E_{21}, \dots, E_{1n}E_{2m}\}$$

E_1 ($E_1 = \{E_{11}, \dots, E_{1n}\}$) and E_2 ($E_2 = \{E_{21}, \dots, E_{2m}\}$) are two sets of entity values. When the two sets are crossed, a new set is generated. It contains all ordered possible pairs (E_{1x}, E_{2y}) where $E_{1x} \in E_1$, $x = 1 \dots n$ and $E_{2y} \in E_2$, $y = 1 \dots m$.

For example, Table 3-4 illustrates the crossing of the “Learner” entity values with the “Activity” entity values, so that users can observe a specific activity (programming or designing) carried out by a specific learner (Tom, Joe or Mike).

Table 3-4 An example of crossing operation

Crossing		E ₁		
		E ₁₁ : Learner(Tom)	E ₁₂ : Learner(Joe)	E ₁₃ : Learner(Mike)
E ₂	E ₂₁ : Activity (programming)	<i>E₁₁ × E₂₁ = Activity: programming ∧ Learner: Tom</i>	E ₁₂ × E ₂₁ = Activity: programming ∧ Learner: Joe	E ₁₃ × E ₂₁ = Activity: programming ∧ Learner: Mike
	E ₂₂ : Activity (designing)	E ₁₁ × E ₂₂ = Activity: designing ∧ Learner: Tom	E ₁₁ × E ₂₂ = Activity: designing ∧ Learner: Joe	E ₁₁ × E ₂₂ = Activity: designing ∧ Learner: Mike

We focus on $E_{11} \times E_{21}$ (the bold italic cell in the above table) to have a close insight into how the traces are manipulated (see Figure 3-11). We define the trace set as a collection of the traces. The trace set T , which includes the integrated traces (IT), is firstly transformed into the trace set T_1 in which the “Activity” attribute of each trace t is programming. Then trace set T_1 is transformed into the trace set T_2 in which the “Learner” attribute of each trace t is Tom. Hence, the crossed trace set T_2 satisfies two crossed entity values (the

activity is programming and the learner is Tom). The other five cells in Table 3-4 are transformed in the same way.

$$\begin{aligned}
 T &= \{t | t \in IT\} \\
 &\downarrow \\
 T_1 &= \{t | \text{Activity} = \text{programming}, t \in T\} \\
 &\downarrow \\
 T_2 &= \{t | \text{Learner} = \text{Tom}, t \in T_1\}
 \end{aligned}$$

Figure 3-11 The crossing process

3.5.1.2 Filtering

The filtering process aims at removing some traces from the integrated traces. The filter process can come after the crossing process. A filtered trace is defined as follow:

$$T = \{t | \text{filter}, t \in \text{crossed trace}\}$$

For example, the learner named Tom wants to observe how he used the chat tool depending on the day. It involves selecting three entities: Learner, Tool and Time. Tom can cross the “Time” entity values with the “Tool” entity values and then add a filter: “Learner=Tom” in order to observe only the traces related to Tom (Table 3-5).

Table 3-5 An example of filter operation

Filtering		E ₁		
		E ₁₁ : Time(01-03-2014)	E ₁₂ : Time(02-03-2014)	E ₁₃ : Time(03-03-2014)
E ₂	E ₂₁ : Tool(chat)	<i>E₁₁ × E₂₁ =</i>	E ₁₂ × E ₂₁ =	E ₁₃ × E ₂₁ =
		<i>Tool: chat</i>	Tool: chat	Tool: chat
		<i>∧ time: 01-03-2014</i>	∧ time: 02-03-2014	∧ time: 03-03-2014
		<i>∧ learner: Tom</i>	∧ learner: Tom	∧ learner: Tom

Figure 3-12 illustrates how the traces are transformed by the filtering process to obtain E₁₁ × E₂₁ (the bold italic cell of the above table). A filter adds a constraint to the crossed traces. In fact, from the crossed trace set T₂ (“Time” attribute is 01-03-2014 and “Tool” attribute is chat), the filter extracts the traces in which the “Learner” attribute is Tom and the new trace set T₃ is generated.

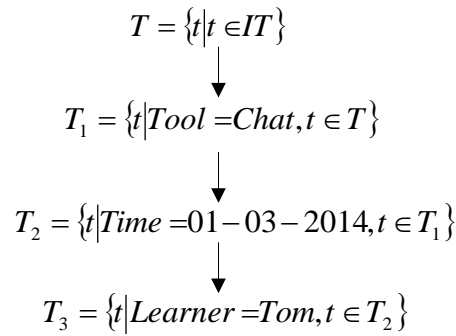


Figure 3-12 The filtering process

3.5.2 Data types

Some data types are used to show the values from the traces that have been crossed and filtered. Five types are proposed: frequency, time interval, time span, content and description.

3.5.2.1 Frequency

This data type can count the quantity of the traces that satisfy two crossed entity values. For example, in Table 3-6, a learner crosses the “Learner” entity values with the “Activity” entity values. If s/he selects frequency as the data type, it means to show how many times each selected learner carried out a specific activity during the project. For instance, the bold italic number in Table 3-6 means that Tom has programmed 2 times.

Table 3-6 An example of the frequency data type

Frequency		E ₁		
		E ₁₁ : Learner(Tom)	E ₁₂ : Learner(Joe)	E ₁₃ : Learner(Mike)
E ₂	E ₂₁ : Activity (programming)	<i>2</i>	3	2
	E ₂₂ : Activity (designing)	15	24	35

Figure 3-13 presents the way to calculate the frequency of programming for Tom (the bold italic number in Table 3-6). After the crossed trace set T_2 (the “Activity” attribute is programming and the “Learner” attribute is Tom) is generated, the number of crossed traces t in the crossed trace set T_2 is counted. The other frequencies are calculated at the same way.

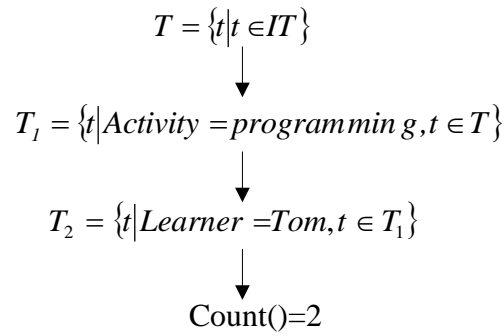


Figure 3-13 The frequency process

3.5.2.2 Time interval

The time interval calculate the total time spent on the two crossed entity values. If we keep the previous example, we obtain the time spent by each selected learner on each activity (Table 3-7). For example, Tom spent 15 hours on programming (the bold italic number in the Table 3-7).

Table 3-7 An example of the time interval data type

Time interval		E ₁		
		E ₁₁ : Learner(Tom)	E ₁₂ : Learner(Joe)	E ₁₃ : Learner(Mike)
E ₂	E ₂₁ : Activity(programming)	<i>15h</i>	10h	1.6h
	E ₂₂ : Activity(designing)	13.2h	17h	20h

Figure 3-14 shows how to calculate the time interval spent by Tom in programming. The crossed trace set T_2 (the “Activity” attribute is programming and the “Learner” attribute is Tom) is generated. The time duration TD of each crossed trace t is the time interval between the “BeginTime” attribute and the corresponding “EndTime” attribute. All the time durations are summed up to generate the total time interval TI .

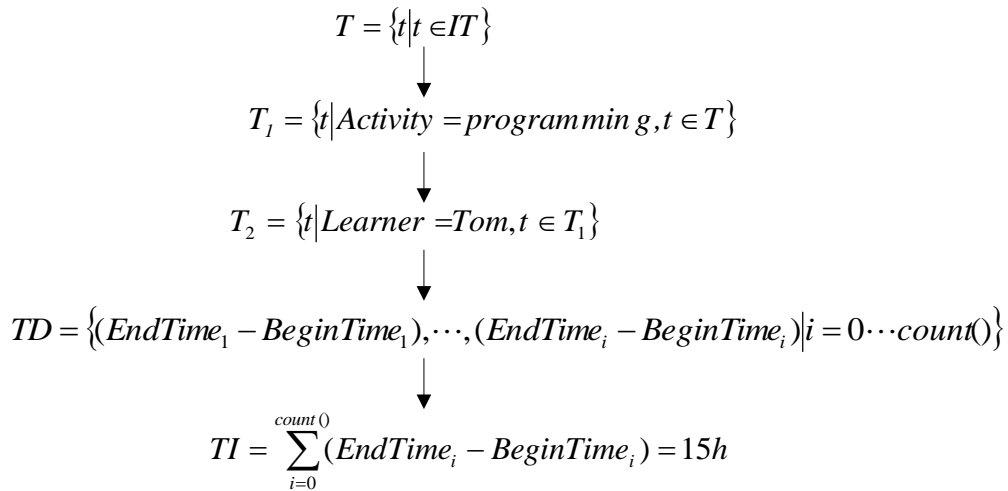


Figure 3-14 The time interval process

3.5.2.3 Content

This data type corresponds to the content of an activity, a reflection, or a judgment, e.g. the content of wiki written by a specific member, the content of the activities carried out by a group member by using a tool (e.g. chat message) or the content of a plan or a judgement. Table 3-8 shows an example. The “Learner” entity values and the “Activity (Plan)” entity values are crossed. It aims at observing the target level of PHP set by each learner in his/her plan report. For example, Tom plans to acquire a high level in PHP (the bold italic word in the Table 3-8).

Table 3-8 An example of the content data type

Content		E ₁		
		E ₁₁ : Learner(Tom)	E ₁₂ : Learner(Joe)	E ₁₃ : Learner(Mike)
E ₂	E ₂₁ : Plan(PHP)	<i>High</i>	Very high	Medium

Figure 3-15 illustrates how the content of the plan about PHP reported by Tom is extracted (the bold italic word in Table 3-8). The crossed trace set T_2 is generated in which the “Learner” attribute is Tom and the “Activity (Plan)” attribute is PHP. Then, the content of each crossed trace t is extracted and generate the result set S , which explains the PHP plan content set by Tom.

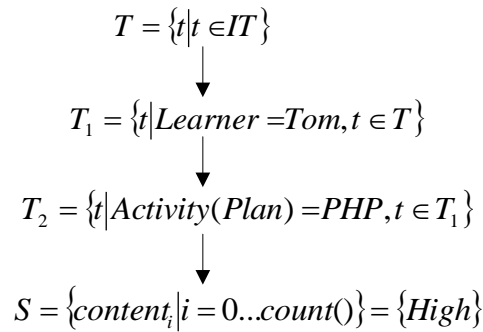


Figure 3-15 The content process

3.5.2.4 Time span

This data type corresponds to both the begin time and the corresponding end time of the learners' activities. An example is given in Table 3-9. It presents the time spans of programming for Tom, Joe and Mike. For instance we can see that Tom programmed from 01-03-2014 10:01:23 to 01-03-2014 10:22:01 and from 05-03-2014 14:00:00 to 05-03-2014 14:50:00 (the bold italic date in Table 3-9).

Table 3-9 An example of the time span data type

Time span		E_1		
		E_{11} : Learner(Tom)	E_{12} : Learner(Joe)	E_{13} : Learner(Mike)
E_2	E_{21} : Activity(programming)	<ul style="list-style-type: none"> • <i>01-03-2014 10:01:23</i> <i>~01-03-2014 10:22:01</i> • <i>05-03-2014 14:00:00</i> <i>~05-03-2014 14:50:00</i> 	<ul style="list-style-type: none"> • 03-03-2014 15:31:03 ~03-03-2014 16:42:31 	<ul style="list-style-type: none"> • 01-03-2014 08:11:43 ~01-03-2014 08:40:51

The Figure 3-16 illustrates how to calculate the time span of the programming activity for Tom. The crossed trace set T_2 (the "Activity" attribute is programming and the "Learner" attribute is Tom) is generated. The begin time and the end time of each crossed trace t is extracted to build a result set TS that shows the time span of programming activity for Tom.

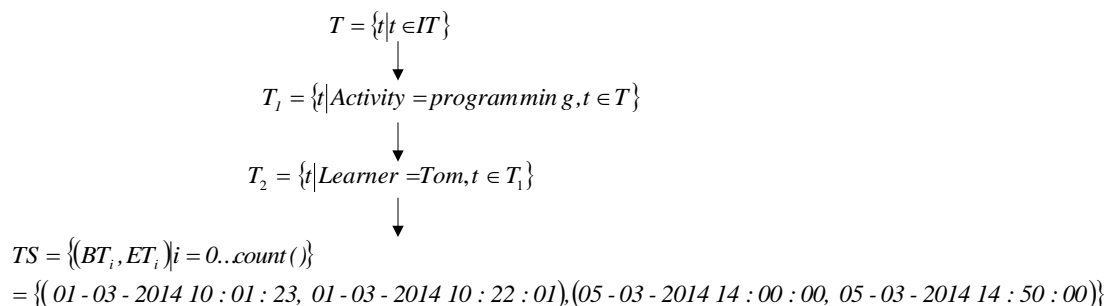


Figure 3-16 The time span process

3.5.2.5 Description

The description is a supplementary data type because it is optional. This option can give more precise and detailed description about the values represented in the indicator. For example, if the project leader wants to not only observe the mood tendencies of all the group members but also find out the reasons of fluctuations, the “content” data type can show their states of minds (high, low, medium) while the description part gives more explanations. These explanations come from the reports written by the learners in the reporting tool. For example in Table 3-10, the bold italic text information below the table gives the explanation of the mood values.

Table 3-10 An example of the description data type

Description (Content)		E ₁		
		E ₁₁ : Learner(Tom)	E ₁₂ : Learner(Joe)	E ₁₃ : Learner(Mike)
E ₂	E ₂₁ : Assess(mood)	<i>High</i>	Very high	Medium

1 information relates to mood: “My mood is high because I solve a problem today”.

Figure 3-17 illustrates how the explanation of Tom’s mood and the statistical information (the bold italic underline word in Table 3-10) are extracted from the integrated traces. The crossed trace set T_2 (the “Learner” attribute is Tom and the “Activity(Assess)” attribute is mood) is generated, and then the crossed traces are classified into small groups according to the “Activity” attribute and the traces of each group are counted. The “Comment” attribute of each trace is extracted and generate the result set S to explain the state of mind of Tom.

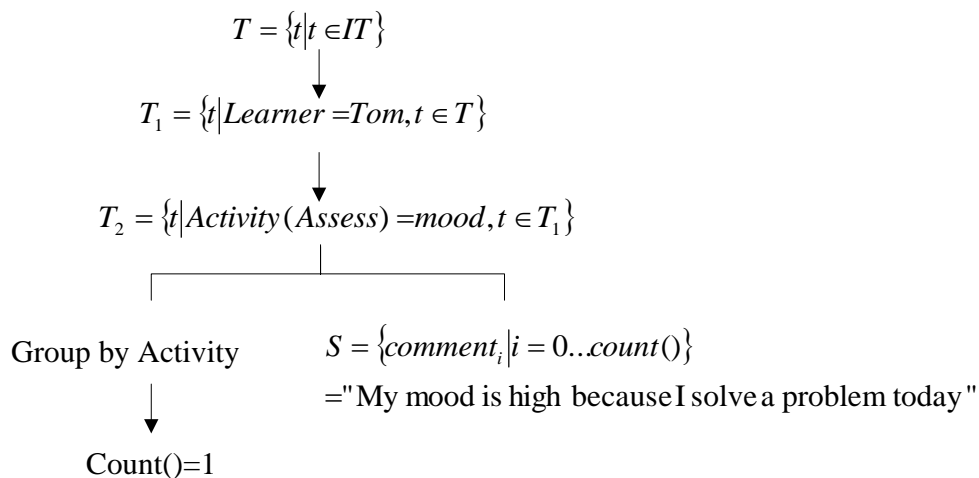


Figure 3-17 The description process

3.5.3 Calculation operators

The calculation operators are mathematic formulas that can be applied on the entity values: addition, subtraction, multiplication and division. According to the complexity, we distinguish two types of operators: simple operators and complex operators. We present these two types of operators in the following parts.

3.5.3.1 Simple operator

A simple operator allows summing up or averaging the results of all the entity values. Table 3-11 illustrates an example that shows the frequencies of the programming and designing activities carried out by Tom and Mike respectively. The column “average” in the table calculates the average frequencies of the programming activity (=1.5) and of the designing activity (=4).

Table 3-11 An example of simple calculation operator

Simple operator (Frequency)		E ₁		average
		E11: Learner(Tom)	E13: Learner(Mike)	
E ₂	E ₂₁ : Activity (programming)	2	1	1.5
	E ₂₂ : Activity (design)	5	3	4

Figure 3-18 explains how the average frequency is calculated of the programming activity (=1.5). The crossed trace sets T_2 (the “Activity” attribute is programming and the “Learner” attribute is Tom) and T_4 (the “Activity” attribute is programming and the “Learner” attribute is Mike) are generated. The crossed traces t in the two sets are counted respectively, then summed up and finally averaged.

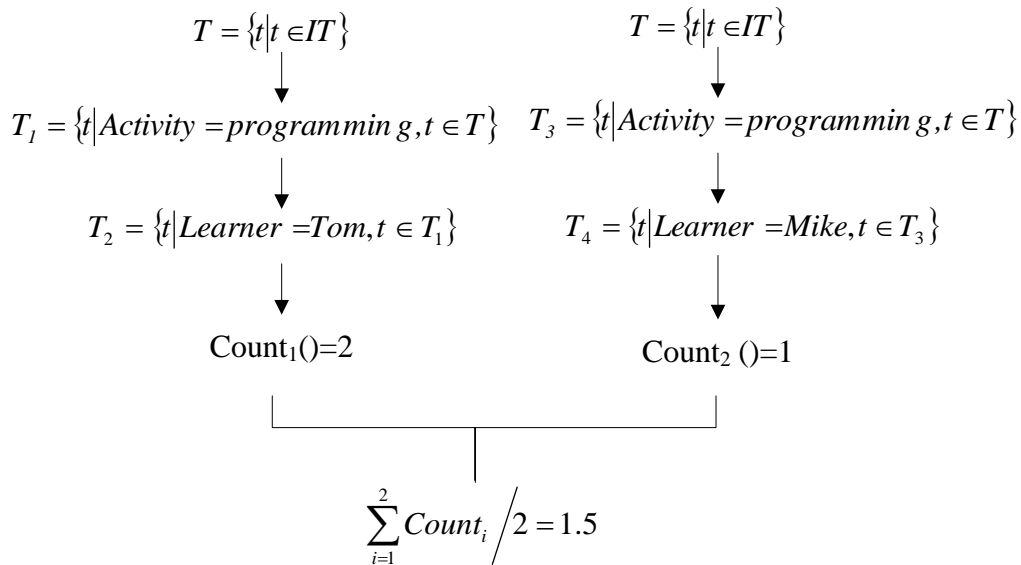


Figure 3-18 The simple operator process

3.5.3.2 Complex operator

Complex calculation operator allows editing complex formula to apply on any two entity values (for example, the column “ $E_{11}+E_{12}$ ” in Table 3-12) or between one entity value and a number (for example, the column “ $(E_{11}+E_{12})/2$ ” in Table 3-12). As illustrated on Table 3-12, we can for example calculate the total frequency of the programming activity carried out by Tom and Joe (=5) and the average frequency (=2.5) (the bold italic number in Table 3-12).

Table 3-12 An example of complex calculation operator

Complex operator (Frequency)		E_1			$E_{11}+$ E_{12}	$(E_{11}+$ $E_{12})/2$
		E_{11} : Learner(Tom)	E_{12} : Learner(Joe)	E_{13} : Learner(Mike)		
E_2	E_{21} : Activity (programming)	2	3	1	5	2.5
	E_{22} : Activity (designing)	5	4	3	9	4.5

Figure 3-19 explains how to perform the complex operator. In order to calculate the total frequency of the programming activity for Tom and Joe, the two crossed trace sets T_2 (the “Activity” attribute is programming and the “Learner” attribute is Tom) and T_4 (the “Activity” attribute is programming and the “Learner” attribute is Joe) are counted respectively. The two frequencies are computed based on the arithmetic operator chosen by the users (in this example the arithmetic operator is +) and then the total frequency is obtained (=5). We

also can calculate the average frequency ($=2.5$) of the programming activity for Tom and Joe by choosing another arithmetic operator $/$ and inputting the number ($=2$).

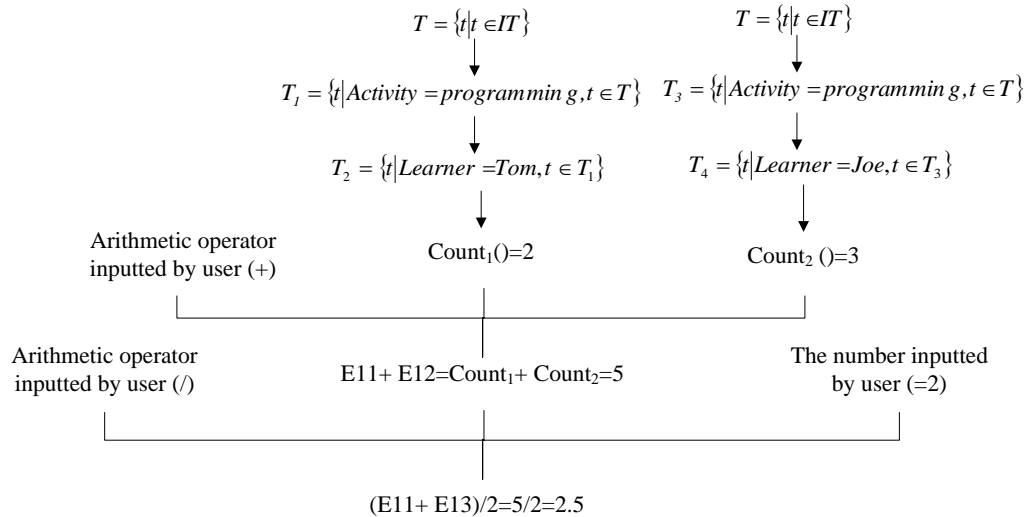


Figure 3-19 The complex calculation operator process

3.6 Indicator visualization

The indicator visualization module presents the indicator calculation results into visually understandable diagrams. The users can choose the visualization dynamically. It is effective to synthesize complex information and so reduce the learners' cognitive load. The input of the visualization process is the indicator results calculated on the parameters described in the previous part (data selection, data types and calculation operators). The process of the indicator visualization module is divided into 3 steps (see Figure 3-20):

1. **Selection of the visualization.** This step is performed by the users. They can select several visualizations to present the indicator. We believe that different visualization modes can help learners to obtain different information 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, scatter charts help to discover the different clusters. Each visualization mode has a unique code, which is used to call the related visualization function in the server side.
2. **Adjustment of the data format.** This step is performed by the system automatically. Considering that each visualization mode has different data format requirements, it is necessary to preprocess the calculation results according to the requirements. The codes of the selected visualization modes (in the previous step) are delivered to the related visualization functions. Then the calculation results can be preprocessed individually according to the required format. If the selected visualization is not suitable for the

calculation results (for example, the calculation result is the time span of several project activities, but the selected visualization mode is bar chart rather than timeline chart), a message is sent to the user to inform that the system fails to adjust the data format.

3. **Indicator presentation.** After the calculation results are adjusted based on the format, they are sent to the corresponding visualization algorithms. The indicator results are presented into different diagrams.

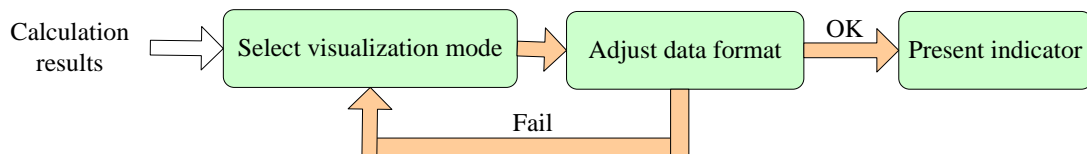


Figure 3-20 The process of indicator visualization

3.7 Conclusion

In this chapter, we propose a PBLMS framework, which aims to help learners to self-regulate in PBL situations. We propose to integrate LMS into PBLMS to supply learners with the collaborative tools to carry out the project. Considering that PBLMS also has to support learners' reflection and self-monitoring processes, we propose to integrate four modules:

1. **A trace collection module.** It collects both activity and reporting traces. We more precisely propose a reporting tool that learners can use to report manually information about their planification, judgment and non-instrumented activities. It supports learners' reflection on the way they carry out the project and collect the non-automatic traceable information. Learners can report the information by filling in semi-structured sentences.
2. **A trace integration module.** We propose a uniform integrated trace model that can integrate the activity traces and reporting traces together.
3. **An indicator calculation module.** We describe the data selection, data types and calculation operators that support to build an indicator based on the integrated traces from a low level to a synthesized level.
4. **An indicator visualization module.** We propose a process to present the calculation results on the form of different diagrams.

In the next chapter, we present an implementation of PBLMS.

4 Implementation of PBLMS

In the previous chapter, we proposed a PBLMS framework from four aspects: trace collection, trace integration, indicator calculation and indicator visualization. This framework supports learners in collaboration, reflection and monitoring on their behaviors and performances during projects. In this chapter, we present an implementation of PBLMS based on the proposed framework. At first, we give an overview of the general architecture. Then, we present the detailed functionalities and associated interfaces and explain the processes that the PBLMS supports.

4.1 General architecture of PBLMS

In the previous chapter, we integrate Learning Management System into PBLMS in order to offer learners a number of collaboration tools to support the communication, information sharing and knowledge construction during the project. We decide to integrate the LMS Moodle¹ into the implementation of PBLMS. Moodle is one of the most popular open-source e-learning platforms, which offers a wide variety of tools (14 default tools and other supplementary tools) to support the teaching and learning processes. It is being used widely around the world, including universities, schools and independent teachers.

We deploy the implementation of PBLMS in the Internet² so it is accessible with a web browser. Learners can access to the PBLMS through HTML pages so that they can visit the system whenever wherever and share the information between group members conveniently. Figure 4-1 is the main interface of PBLMS. In the center of the page, some tools that can be used by the learners are listed. These tools can be divided into two parts:

1. The last four tools (Chat, Forum, Wiki, Resource) are coming from Moodle and can support collaborations in PBL. Learners can access to these tools to communicate, share information, co-construct knowledge during the project. Figure 4-2 presents the main interfaces of these Moodle tools.
2. The Reporting tool (specified in section 4.2) and Dashboard (specified in section 4.3), which are named as DDART (Dynamic Dashboard based on Activity and Reporting Traces), are developed to support reflection and monitoring functions. DDART enables learners to complete and monitor their projects by proposing two dynamic aspects. On the one side, it allows learners to input information that cannot be captured automatically by the system so as to get a more complete view on learners' performances, reflections and activities during the project. On the other side, it supports learners to self-monitor by creating

¹ <https://moodle.org/>

² <http://ddart.fr/moodle>

customizable indicators in a WYSIWYG (What You See Is What You Get) way.

Hence, the implementation of PBLMS that we propose in this chapter is composed of Moodle and DDART.

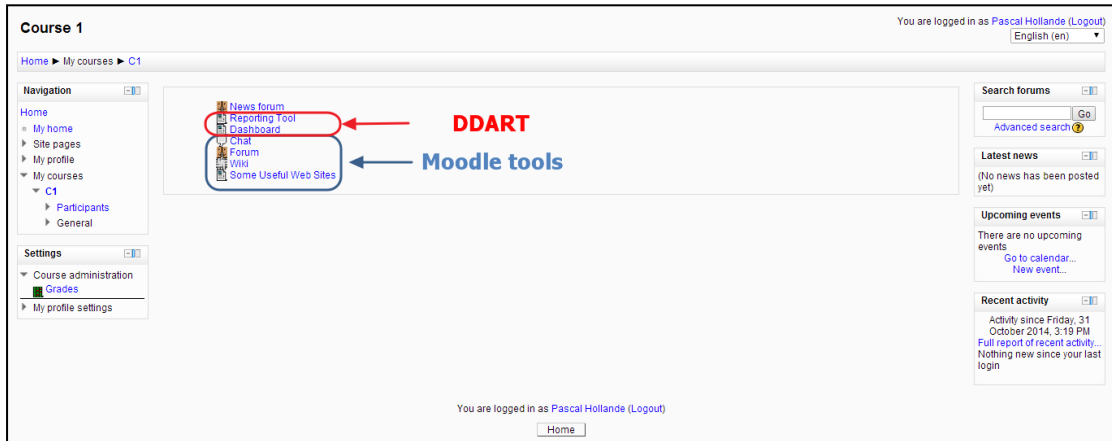


Figure 4-1. The main interface of PBLMS

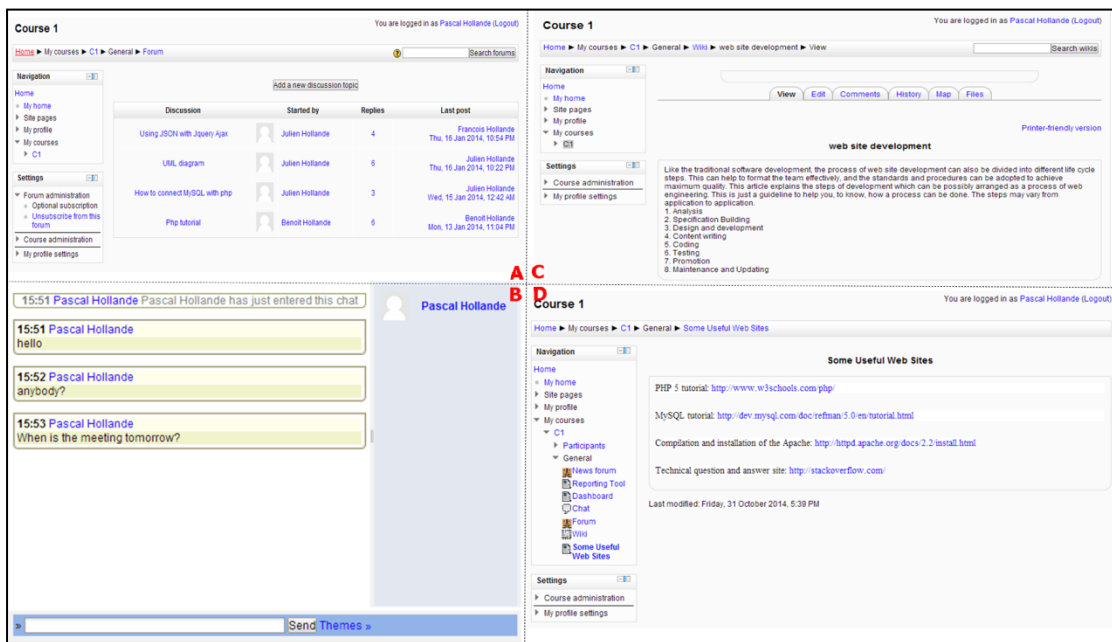


Figure 4-2 The main interfaces of the Moodle tools
(A) Forum, (B) Chat, (C) Wiki, (D) Resource

The general architecture of DDART is composed of five processes (see Figure 4-3): trace collection, trace integration, indicator calculation, indicator visualization, and indicator management. These processes enable learners to explore the activity traces and reporting traces by creating customizable indicators so as to follow the project and their behaviors globally.

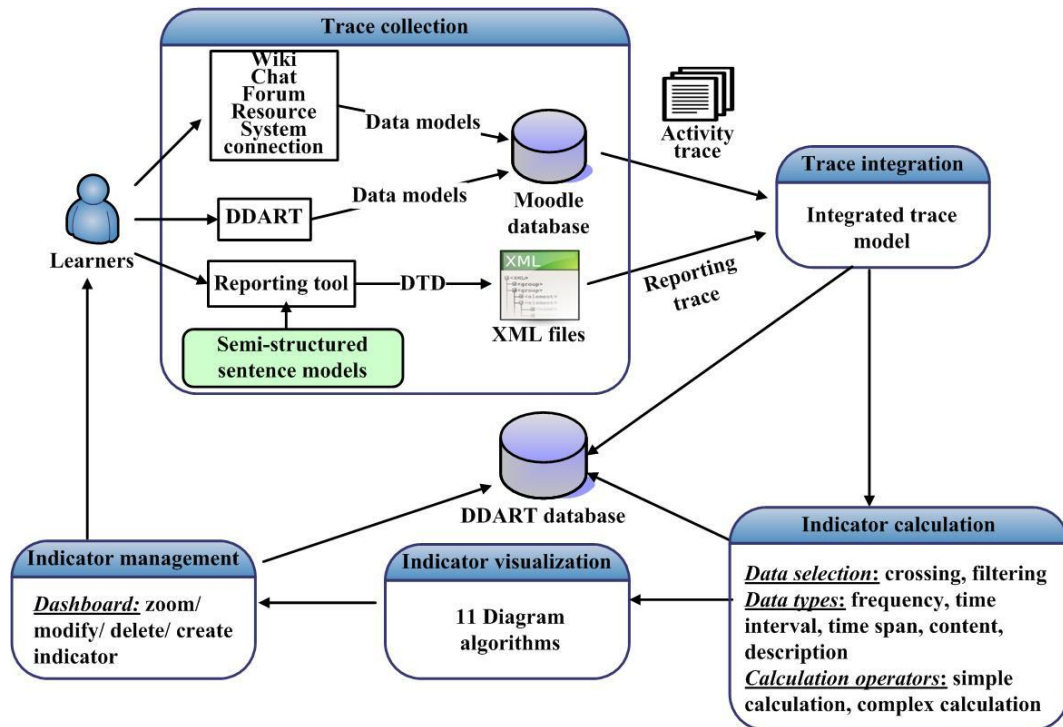


Figure 4-3 The architecture of DDART

Trace collection. This process is used to collect the activity traces and reporting traces. In order to collect the activity traces (see Figure 4-4), we decide to focus on the activities performed with four default tools in Moodle (Wiki, Chat, Forum and Resource), the users connections to the system, as well as the activities performed with two proposed tools in DDART: the reporting tool and the dynamic dashboard. Considering that Moodle enables to track the activities traces generated in the four default tools and the system connections, we decide to reuse the traces recorded by Moodle. These activity traces are stored in the Moodle database based on the data models (Annex A). We have to define the tracking of DDART tools (write report, read report, comment report, view dashboard, create indicator) and so we have developed the specific sensors (in JavaScript and AJAX which handles the data asynchronously) on the browser side to capture these interactions. When learners carry out the specified activities in the browser side, the collect sensors can be triggered and send the information to the server side to be processed. The server side uses the corresponding data models (Annex B) to store the information so as to generate the corresponding activity traces. This function is developed in PHP, which is a widely-used and open source scripting language. We chose MySQL³ as the database management system, which is also used in Moodle, because it is more convenient to reuse the activity traces in Moodle database. This database is also used to store and calculate the integrated traces and manage the indicators later.

³ <http://www.mysql.com/>

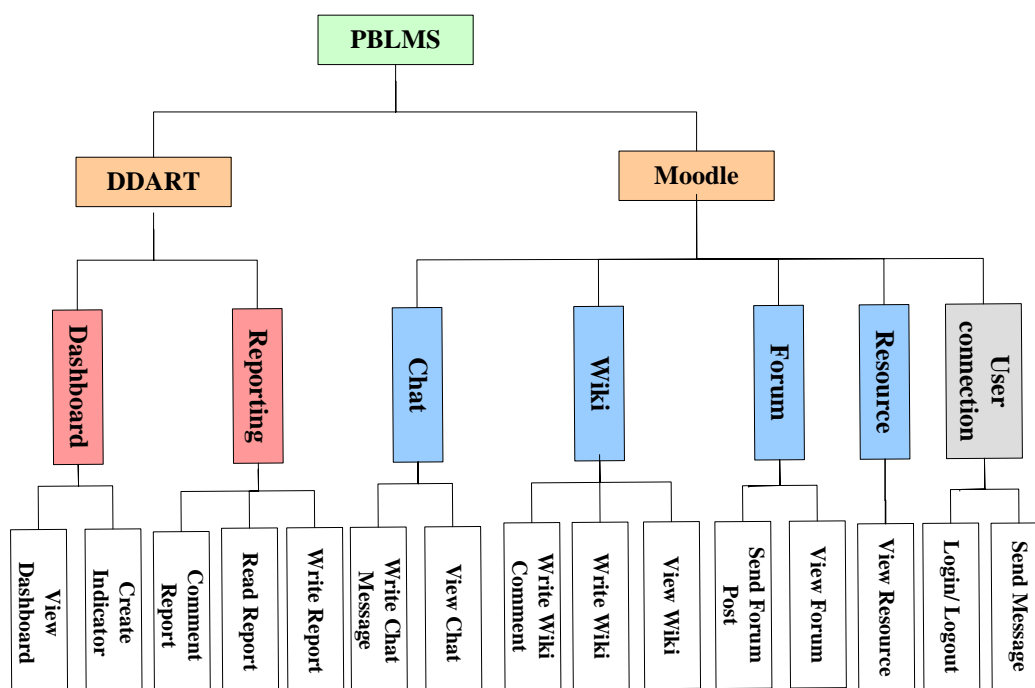


Figure 4-4 The automatic traceable activities in PBLMS

For the reporting traces, learners use a specific reporting tool to record their project activities carried out without PBLMS. This tool is based on the semi-structured sentence models (described in Table 4-1 in section 4.2). According to the class models of reporting traces (Figure 3-9) that we propose in previous chapter, we design the DTD (Document Type Definition) for each class model (Annex C). DTD structures the data by defining the legal building blocks of an XML document. The reports are saved as XML files based on the DTD. We select BaseX⁴ to act as the database management system. It is a pure xml database and support of the W3C XPath/XQuery.

Trace integration. This process is used to formalize and integrate the reporting traces and activity traces together according to the integrated trace model (see section 3.4) we propose in previous chapter. The integration process is called automatically when learners access to the dashboard or create customizable indicators. This process ensures that the results of indicators are updated. The integrated traces are stored in the MySQL database.

Indicator calculation. This process enables learners to design customizable indicators by manipulating the integrated traces. Learners can apply data selection (crossing, filtering), data types (frequency, time interval, time span, content, description) and calculation operators (simple operator and complex operator). These methods allow learners to focus on particular entity values of the traces and aggregate the traces by different methods. Learners can create plenty of customizable indicators to compare his/her own activities with the other group members and at different time periods. The indicator is defined

⁴ <http://basex.org/>

by some parameters, such as the observed entity values, the data types and the calculation operators. All these parameters are supplemented with basic information on the indicator, such as indicator name, creator's name, description, and are stored in the database.

Indicator visualization. This process supplies eleven visualization modes to present the indicator results. Learners can select several modes to visualize the indicator from different points of views.

Indicator management. The indicators created by a learner are presented in his/her personal dashboard. Learners can monitor and reflect on their own project performances by observing these indicators. The dashboard offers the functions of indicator management, for example, indicator deletion, zoom, modification and creation.

In the next parts, we illustrate the tools of DDART: the reporting tool and the dynamic dashboard. The interfaces are presented to give a graphical interpretation and the functions are explained and illustrated with examples.

4.2 Reporting tool

The objectives of the reporting tool are 1) to help learners to reflect on their project activities and learning and 2) to collect information of the activities which cannot be traced automatically by the system, such as the project plans, the project activities carried out without PBLMS, the peer-judgments, 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.

Figure 4-5 illustrates the interface to write a new report. This functionality allows learners to write a new report to record their project plans, activities and reflections. Learners have to write the report title and select the report visibility (individual: the report cannot be viewed by others, public: the report can be read by others). They can click the left blue button at the bottom of the interface to add some semi-structured sentence models (Table 4-1) into the report. Learners have to select the models that they want to declare from the message box (see Figure 4-6a) by checking the corresponding checkboxes. They can fill in the selected semi-structured sentences in the editor area (see Figure 4-6b) based on their own situations or delete some selected sentences from the editor area or add more semi-structured sentences. The data will be stored in the xml database BaseX when learners submit the report.

Figure 4-5 The interface of writing a new report

(a)

(b)

Figure 4-6 The interfaces of reporting tool

(a) Semi-structured sentence models (b) The selected sentences in the editor area

According to the categories of semi-structured sentences proposed in the last chapter (see section 3.3.2), we propose several models based on different report types (see Table 4-1). These models can lead learners to reflect and encourage them to record the useful information that they are able to explore further.

Table 4-1 The semi-structured sentence models

Report type	Semi-structured sentence models	Description
Goal report	The project goal is to <i>do what</i> from <i>time</i> to <i>time</i> .	Learners have to describe the project objective and set the time period.
	My <i>skill/knowledge</i> should reach the level of <i>very good/ good/ neutral/ not very good/ not at all</i>	Learners set the ideal levels (very good/ good/ neutral/ not very good/ not at all good) of a specific skill (e.g.

	<i>good.</i>	communication, documentation) or a particular knowledge (e.g. PHP, database) that they want to arrive after the project.
	I schedule to <i>do what</i> from <i>time</i> to <i>time</i> .	Learners decompose the project into small activities and set the schedules for these activities by stating the time period.
Activity report	I <i>do what</i> with <i>whom</i> about <i>what</i> by using <i>what tool</i> from <i>time</i> to <i>time at place</i> .	Learners record an activity carried out outside of PBLMS by declaring the activity elements: do what, with whom, about what, how, when and where.
	I judge that <i>what</i> is <i>very good/ good/ neutral/ not very good/ not at all good</i> because <i>reason</i> .	Learners announce a judgment of a particular group peer or an object (e.g. a book, a web site) and give the reasons.
	I self assess that my <i>skill/ knowledge/ mood...</i> is <i>very good/ good/ neutral/ not very good/ not at all good</i> because <i>reason</i> .	Learners assess one of their skills or knowledge or affective states and give the reasons.

The functionality named as “Manage my reports” enables learners to modify the contents of their reports and to reply to the comments written by the other group members. Figure 4-7 presents the list of the reports written by the learner. When s/he selects one of the reports that s/he wants to update, the content and comments of this report will be showed below the report list (see Figure 4-8 and Figure 4-9).













Write a new report	Manage my reports	Read other reports																				
<table border="1"> <thead> <tr> <th>Title</th> <th>Date</th> <th>Privacy</th> <th>Edit</th> </tr> </thead> <tbody> <tr> <td>my goal</td> <td>12-1-2014 23:12:44</td> <td>public</td> <td></td> </tr> <tr> <td>12/1 activity report</td> <td>12-1-2014 23:42:42</td> <td>public</td> <td></td> </tr> <tr> <td>13-1 report</td> <td>13-1-2014 23:24:27</td> <td>public</td> <td></td> </tr> <tr> <td>14-1 report</td> <td>14-1-2014 23:42:20</td> <td>public</td> <td></td> </tr> </tbody> </table>			Title	Date	Privacy	Edit	my goal	12-1-2014 23:12:44	public		12/1 activity report	12-1-2014 23:42:42	public		13-1 report	13-1-2014 23:24:27	public		14-1 report	14-1-2014 23:42:20	public	
Title	Date	Privacy	Edit																			
my goal	12-1-2014 23:12:44	public																				
12/1 activity report	12-1-2014 23:42:42	public																				
13-1 report	13-1-2014 23:24:27	public																				
14-1 report	14-1-2014 23:42:20	public																				
2 pages(1/2) [1][2]																						

Figure 4-7 The report list written by a specific group member

The learner can select a specific report, modify its contents and then save the modifications (Figure 4-8). It can be useful for instance if s/he thinks that the information in a report is incorrect or not complete.

Report content

I organize a meeting with benoit,sophie,francois,p about distribution of work using Skype from 11-01-2014 09:00 to 11-01-2014 09:30 at home .

I judge that benoit is good because he is active in the meetir .

I self assess that my mood is very good because this project is interesting .

save modification

Figure 4-8 Modification of the report contents

Figure 4-9 illustrates the functionality of replying to the comments written by the other group members on a specific report. The learner can reply directly (see upper part of Figure 4-9) or write a new comment (see bottom part of Figure 4-9). This comment function can improve the communication between learners and enable the learner to know how the others view his/her works.

Comments

1). benoit:
good report!

Reply the comment here

Write some comments here

Reply to a comment

Upload the comment

Add a new comment

Figure 4-9 To reply the comments related to a report written by the learner

The reporting tool supplies another functionality that allows learners to read the reports written by other group members. It encourages learners to share their experiences and help them to follow the progresses of the other group members. Learners can select one report whose visibility is public (see Figure 4-10) to visualize the report contents and the associated comments and write some new comments (see Figure 4-11).

Write a new report Manage my reports **Read other reports**

Title	Date	Writer	Detail
my goal report	12-1-2014 22:33:46	Benoit	
first activity report	12-1-2014 22:55:45	Benoit	
my goal	12-1-2014 23:12:44	Pascal	
12/1 activity report	12-1-2014 23:42:42	Pascal	
project goal	12-1-2014 23:43:13	Francois	

5 pages(1/5) [1][2] [3] [4] [5]

Request report detail

Figure 4-10 The report list written by other group members

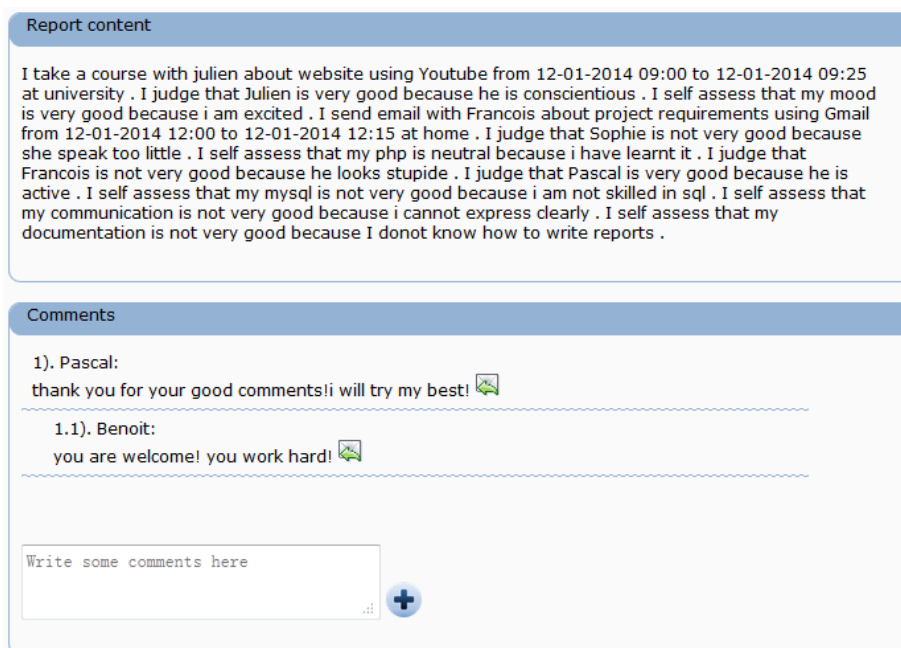


Figure 4-11 The content and comments of a report written by another group member

4.3 Dynamic dashboard

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

All the indicators that have been created by learners are presented in their own dashboard. For instance, in Figure 4-12, the dashboard is composed of three indicators. Each indicator is consisted of three parts: the indicator name, the indicator description and the graphical part. The learners can manage the indicators by zooming, updating or deleting. They can observe these indicators separately or compare them to discover the problems in their project 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.

The three indicators presented in the dashboard in Figure 4-12 focus on the different aspects. The first indicator “The judgments 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 hours) on the project at 12th Jan 2014 and the average working time is 4.3 hours from 11th Jan 2014 to 16th Jan 2014. The third indicator “Social network of our group” presents the interaction frequencies between three group members: Sophie, Francois and Julien. We can observe that the Julien contacts Francois frequently while Sophie seldom contacts her teammates.

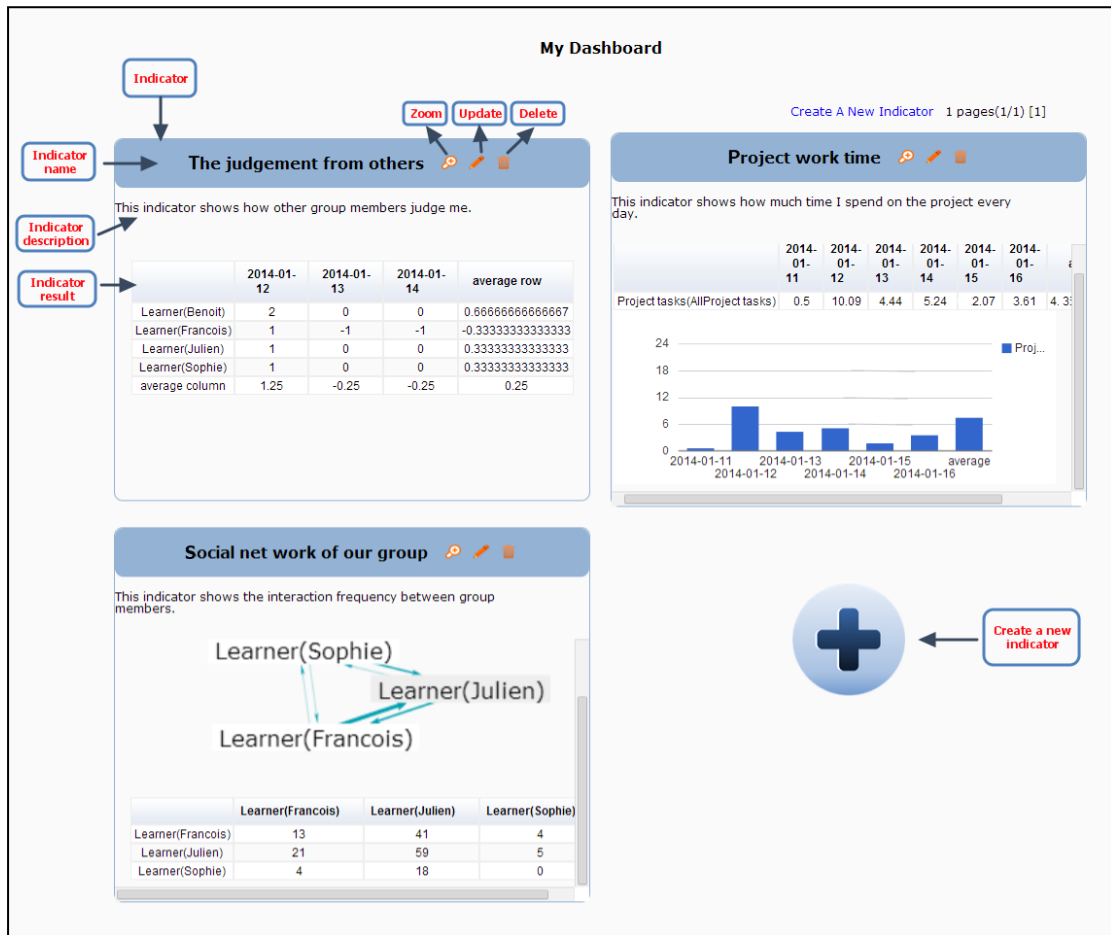


Figure 4-12 The dashboard interface

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 a new interface dedicated to the indicator creation or modification.

Figure 4-13 illustrates the interface to create a new indicator, which can be divided into five parts.

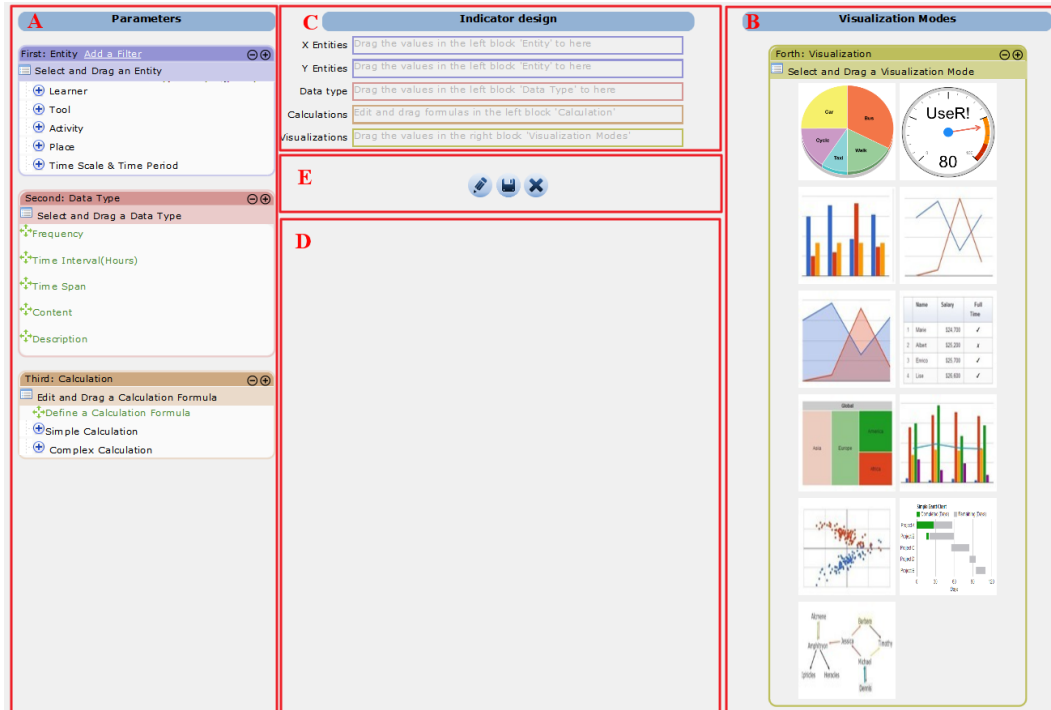


Figure 4-13 The interface to create an indicator

- (A) The “Parameters” part, on the left side of the page (Figure 4-13), contains the list of all the parameters which are available for creating an indicator, including the “Entity” block (see Figure 4-14a), the “Data type” block (see Figure 4-14b) and the “Calculation” block (see Figure 4-14c).

The “Entity” block contains five types of entities: Learner, Tool, Activity, Place, Time scale and time period. We have introduced these five entities in the previous chapter (see section 3.4), which are the main elements extracted from the integrated traces. Under each entity, some corresponding values are listed:

1. Under the “Learner” entity, all the group members’ names are listed.
2. Under the “Tool” entity, all the tools in PBLMS (Wiki, Forum, Chat, Resource, Report, Dashboard) and the other tools written in the reporting tool by learners (e.g. MS Word, Skype etc.) are listed.
3. Under the “Activity” entity, there are three sub-categories. The first one “Plan” contains all the objects of learners’ plans (e.g. the name of a skill or knowledge, the project goals etc.), the second category “Project tasks” contains all the project activities carried out inside of PBLMS (e.g. send a forum post, write reports, edit wiki etc.) or outside of PBLMS (e.g. search information in the internet, brainstorm, face to face discuss etc.) and the third category “Judge” contains all the objects of learners’ judgments (e.g. a book, a group peer, the mood etc.).
4. Under the “Place” entity, all the places where the activities are carried out are listed.

- Under the “Time scale and time period” entity, time scale is divided into three granularities: day, week and month. Learners can select one of the granularities to aggregate the traces. They also can specify the time period by setting the beginning time and the end time.

Learners have to select the specific values that they want to explore from the “Entity” block and define how to cross these selected items.

The “Data type” block lists all the data types (Frequency, Time interval, Time span, Content and Description) we introduce in previous chapter (see section 3.5.2). Learners have to select one of the first four types to explore the different aspects of the selected crossed entity values. “Description” is a special data type because it is optional. It can offer more supplementary information about the indicator results.

The “Calculation” block supplies calculation operators to enable learners to calculate the indicator results further. It is divided into two parts: simple calculation and complex calculation. In the simple calculation, learners can calculate the sum or the average value of all the indicator results at one time. In the complex calculation, learners can do the calculation between any two entities values or between one entity value and one number.

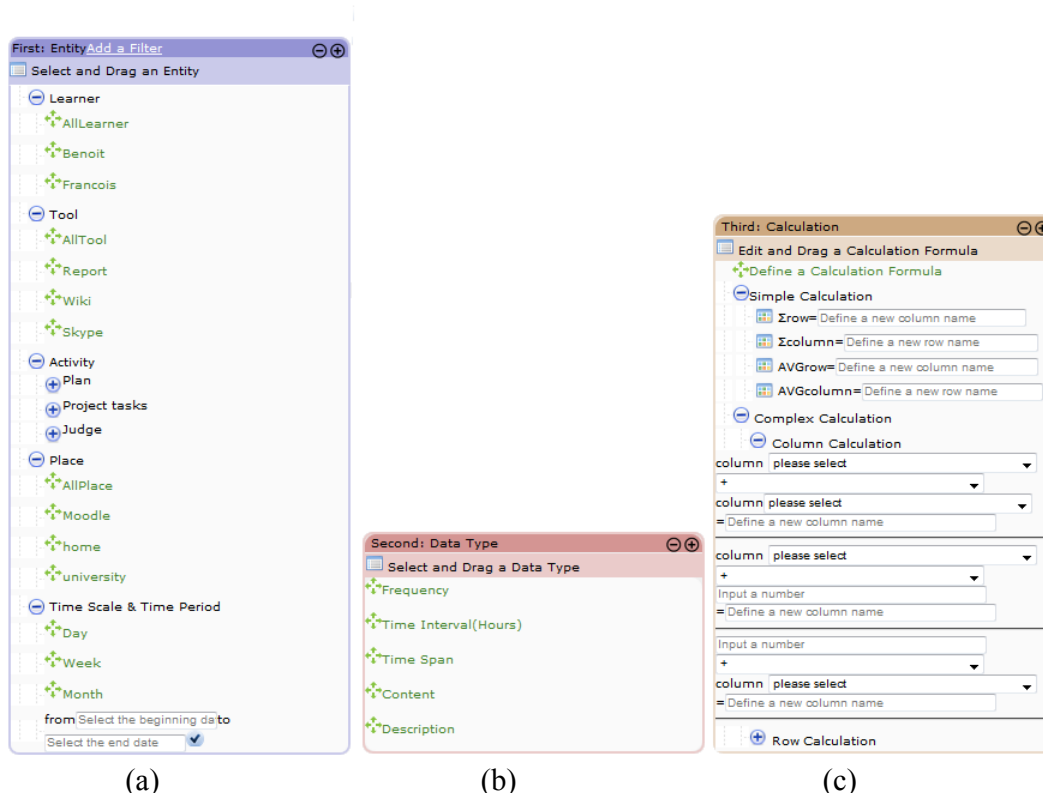


Figure 4-14 The parameter blocks in the interface to create an indicator
 (a) The “Entity” block, (b) The “Data type” block, (c) The “Calculation” block.

(B) The “Visualization modes” part, on the right side of the interface (Figure 4-13), supplies eleven visualization modes. In Annex D, we give the examples of each mode. Learners can select one or several modes to visualize the indicator results from different points of views.

1. **Table.** It arranges the data in columns and rows, which can present the precise numerical data and the text data. Table is the default visualization mode of the indicator.
2. **Gauge chart.** A gauge chart contains values in a dial which is cut into several segments (generally, the segments are low, medium and high). Learners can know from the gauge chart in which segment the value is.
3. **Pie chart.** It is divided into several segments and shows the numerical proportions of each segment of a whole. Learners can compare the proportions between different segments.
4. **Bar chart.** This graph shows the differences among categories in individual values horizontally. Learners can compare the different values of several categories. But all the values are required to base on one measurement unit.
5. **Gantt chart.** It illustrates a project schedule by describing the start and finish dates of each project activity plan or depicts how a set of project activities are carried out over time in real.
6. **Combo chart.** It lets users to combine different types of chart in one chart by rendering each category as a different marker type: line, area, bars, candlesticks, and stepped area. It is useful for learners to compare values between different categories.
7. **Line chart.** This graph shows the fluctuations of values. Learners can compare the general trends of different lines or compare the values of different lines at a particular time point.
8. **Area chart.** This graph is based on line chart. It shows the proportions of different parts to a whole over time and learners can compare the values by areas.
9. **Scatter plot.** It presents the plot points on a graph, which helps learners to distinguish the clusters.
10. **Tree map chart.** It shows a data tree. Each tree node can have zero or more children and is displayed as a rectangle. The size and the color of the tree node are based on the values.
11. **Network chart.** It shows the relationships between different elements. The line weights between any two elements are calculated based on the values.

We import the Google Visualization API⁵ to generate the first ten visualization modes and import Springy⁶, which is a force directed graph layout algorithm, to produce the network chart.

⁵ <https://developers.google.com/chart>

⁶ <http://getspringy.com/>

- (C) The “Indicator design” part, in the top center of the interface (see Figure 4-13), allows learners to drag the indicator parameters from the “Parameters” part and the “Visualization modes” part and drop them in the corresponding panes. There are five panes in the “Indicator design” part for placing the different parameters: the “X Entities” pane, the “Y Entities” pane, the “Data type” pane, the “Calculations” pane and the “Visualizations” pane. The five parameters can compose an indicator. Figure 4-15 gives an example of creating an indicator. This indicator is built to observe the usage frequencies of “Chat” tool and the “NetBean” tool used by Pascal from 13-01-2014 0:00:00 to 16-01-2014 0:00:00. Next, we explain the functions by illustrating how to create this indicator.

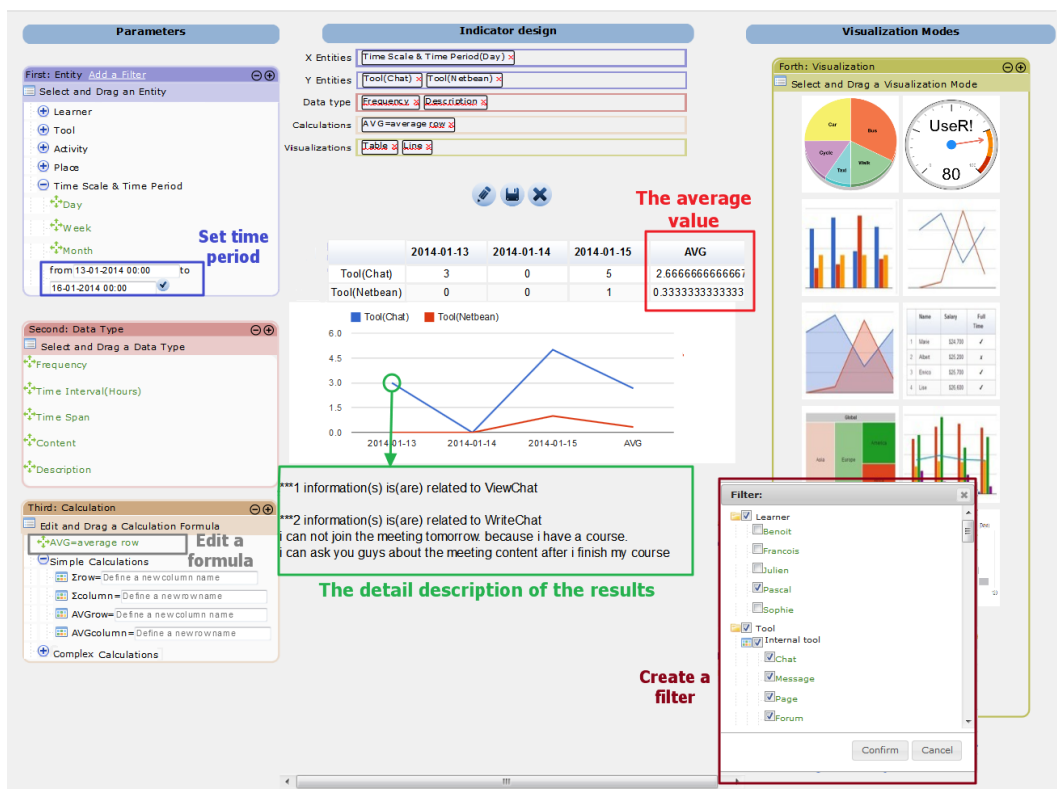


Figure 4-15 The example of creating indicator

The “X Entities” pane and “Y Entities” pane are used to drop the entity values selected from the “Entity” block. Each value placed in the “X Entities” pane can cross with each value in the “Y Entities” pane. In Figure 4-15, the entity value “Time scale & time period (day)” is listed in the “X Entities” pane and the entity values “Tool (Chat)” and “Tool (NetBean)” are dropped in the “Y Entities” pane. Hence, the entity value “Time scale & time period (day)” is crossed with the entity values “Tool (Chat)” and “Tool (NetBean)” respectively. The time period is set in the “Entity” block (highlighted with the left top frame in the Figure 4-15).

The filter message box helps learners to filter the traces if learners want to narrow the data further. Learners can uncheck the entity values listed in the message box which they do not want to explore. For example, in Figure 4-15, the indicator is related to the information on the tools used by the learner Pascal during a period. Hence, a filter is necessary to exclude the traces which are not related to Pascal from the database. The filter (highlighted with the right bottom frame in the Figure 4-15) can be created by unchecking other members' names and keeping the name Pascal checked.

Learners can drag a data type from "Data type" block and drop it in the "Data type" pane. The indicator values can be calculated according to the selected data type. In Figure 4-15, 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. The frequencies can be calculated immediately. 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 of each tool every day, they can drag the data type of "Time Span"; and if they want to know the interaction content of each tool every day (chat content, wiki content...), they can drag the data type of "Content". In the example presented in Figure 4-15, 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 4-15).

The "Calculations" pane is to place the mathematics formulas edited by learners in the "Calculation" block. After learners edit the formula in the "Calculation" block, they can drag and drop it in the "Calculations" pane. In the example, in order to calculate the average frequency of each tool, learners need to edit a formula (AVG= average row) in the "Calculation" block and drag it to the "Calculations" pane (highlighted with the left bottom frame in Figure 4-15). The average values can be calculated and presented (highlighted with top right frame in Figure 4-15).

Learners can drag any appropriate visualization modes and drop them in the "Visualizations" pane. The indicator will be then visualized in the selected modes. In this example, the visualizations of line chart and table are selected. Hence the indicator is presented in these two forms. The "Table" visualization mode presents the precise results while the line chart is useful to observe the trend of usage frequency of each tool according to the date.

- (D) The "Result" part, in the bottom center of the Figure 4-15, is used to present the created indicator. The result is calculated in real time. When a parameter

is added, deleted or updated, the indicator is update accordingly. In this way, learners can follow the indicator calculation and get the results quickly.

- (E) The “Menu” part, in the middle centre of Figure 4-15, contains some icons which allow learners to manage indicators (e.g. updating an indicator, saving an indicator, deleting an indicator and canceling the actions). When learners want to update an indicator, they need to select a specific indicator. After they complete the modification, if they are not the first creator of this indicator, they have no right to modify it and they just can save it as a new own indicator. If they are the original creator of this indicator, they can select to replace the old indicator or save it as a new one.

Learners can save the indicator and name it (Figure 4-16). They need to give a name, set the visualization scope and write some descriptions for this indicator. If the indicator is saved as public, it is shared with the other group members. If it is individual, it cannot be reused or seen by others. After the indicator is created, it is presented in the dashboard.

Figure 4-16 the interface to save an indicator

In order to manage the indicators easily, we store them in the MySQL database, which allows making SQL queries. Table 4-2 is the data dictionary of the indicator and contains the information related to an indicator.

Table 4-2 Data dictionary of an indicator

Data field	Type	Description
Id	Int (auto_increment)	The indicator id.
Name	String	The name of indicator.
Privacy	String	The indicator is public or private.
Description	String	The description of the indicator written by

		the creator.
Creator id	String	The creator id.
Create time	DateTime	The creation time.
Module id	Int	The module id of the dashboard.
Entity X	String	The X entity values
Entity Y	String	The Y entity values to be crossed with the values in Entity X
Filter	String	The filter of indicator.
Data type	String	The data type of the indicator
Mathematic formula	String	The mathematic formula of the indicator.
Visualization modes	String	The visualization modes of indicator.

4.4 Conclusion

In this chapter, we illustrated an implementation of the PBLMS framework proposed in the third chapter. We integrated the LMS Moodle into PBLMS to supply collaboration tools (Chat, Forum, Wiki, Resource) for learners to carry out the project. We also developed a system, named DDART, which can help learners to reflect on their activities and monitor themselves by creating customizable indicators. DDART is composed of a reporting tool and a dynamic dashboard. The reporting tool aims to help learners to reflect on their own performances as well as to collect the reporting traces. The dynamic dashboard enables learners to create customizable indicators by exploring the activity and reporting traces. This dashboard can support them in the monitoring of their own learning and project progresses.

In the next chapter, we validate DDART by building various indicators proposed in the literature to verify the genericity of the implemented processes. Furthermore, we present the results of an experiment we conducted with real learners in order to test the usability and utility of DDART.

5 Illustration of use, experiment and result analysis

In the previous chapter, we presented an implementation of PBLMS. In this chapter, we test the performance of the system by using two ways:

- 1 The creation with DDART of a large sample of indicators that are proposed in existing researches about the analysis of activities, cognition, emotion and social network. We can so evaluate the possibilities offered by the features of DDART for the creation of various indicators and the limitations of this system.
- 2 An experiment in order to test the utility and usability of DDART with end users. Their feedbacks and experiment results are helpful to improve the functionalities of DDART.

At the end, we draw the conclusions of the experiment and give a broader perspective.

5.1 Test of the ability of DDART to create existing indicators

In the chapter 3, we propose several categories of indicators according to their different dimensions: activity indicators, social indicators, cognitive indicators and affective indicators. In order to verify the features of DDART, we create indicators coming from these four types and already proposed in existing 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 has five members: Sophie, Julien, Pascal, Benoit and François. The project begins from 11, Jan 2014 to 16, Jan 2014. 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.

5.1.1 Activity indicators

5.1.1.1 Students login overview

Mazza and Milani (2004) proposed a graphical interactive student monitoring tool called GISMO. Figure 5-1 presents an indicator which shows the learners' accesses to the system. By observing this indicator, group leaders or tutors can have a general view of all accesses performed by each group member and obtain a clear identification of patterns and trends at a glance. In the upper part, the access matrix lists the learners' names (on Y axis) and the dates of the accesses (on X axis). The corresponding red square represents at least one access to the

system made by the group member on the selected date. On the bottom, the histogram shows the total accesses to the system by all the learners on each date.

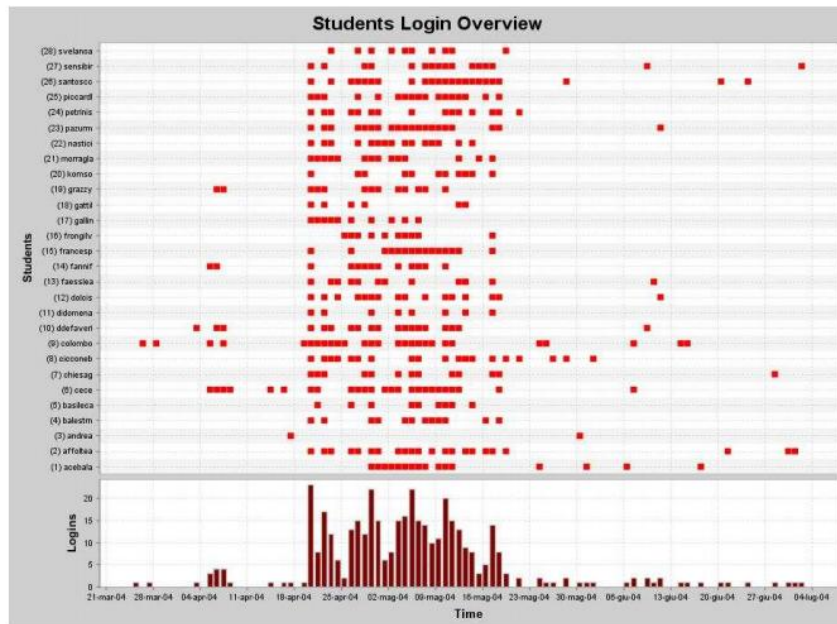


Figure 5-1 The visualization of “students login overview” indicator (Mazza and Milani 2004)

Figure 5-2 is the main interface of creating the above indicator by using the dashboard of DDART. The indicator parameters are listed in the “Indicator design” part. The “Project tasks (Login)” is a value of the “Activity” entity. It is crossed with “Learner (Benoit)”, “Learner (Francois)”, “Learner (Pascal)”, “Learner (Sophie)” and “Learner (Julien)” respectively, which are the values of the “Learner” entity. The data type of the indicator is “Time Span” which presents the access time slots performed by each learner. The “Gantt” visualization enables the indicator to be presented in a timeline format. The time period of the indicator is set from 11-01-2014 00:00 to 15-01-2014 00:00 in the “Entity” block. The indicator results are calculated based on the parameters and are showed in the “Result” part.

The screenshot shows the 'Indicator design' interface. On the left, the 'Parameters' section is active, showing three configuration blocks: 'First: Entity Add a Filter' with a tree view of entities (Learner, Tool, Activity, Place, Time Scale & Time Period) and a date range from 11-01-2014 00:00 to 15-01-2014 00:00; 'Second: Data Type' with options like Frequency, Time Interval(Hours), Time Span, Content, and Description; and 'Third: Calculation' with options for Simple and Complex Calculations. The 'Indicator design' section on the right shows 'X Entities' as 'Project tasks(Login)', 'Y Entities' as five learner names, 'Data type' as 'Time Span', and 'Visualizations' as 'Gantt'. Below this is a Gantt chart with five rows for learners and two columns for 'Mon' and 'Tue', with vertical bars indicating login activity.

Figure 5-2 The interface to create the “students login overview” indicator (1)

Figure 5-3 illustrates the calculation of the daily access frequencies to the system performed by all learners. The time period is set from 11-01-2014 00:00 to 15-01-2014 00:00 in the “Entity” block. The indicator parameters are listed in the “Indicator design” part. The “Time scale & Time period (Day)” entity value is listed in the “X Entities” pane, which is crossed with the “Project tasks (Login)” entity value listed in “Y Entities” pane. The data type is “Frequency”. These three parameters are used to calculate the daily login frequency. The indicator visualization is “Table” and “Bar” chart. The indicator result is calculated based on the parameters and is showed in the “Result” part. By observing the Figure 5-3, we can know that the highest access frequency occurs in 13-01-2014.

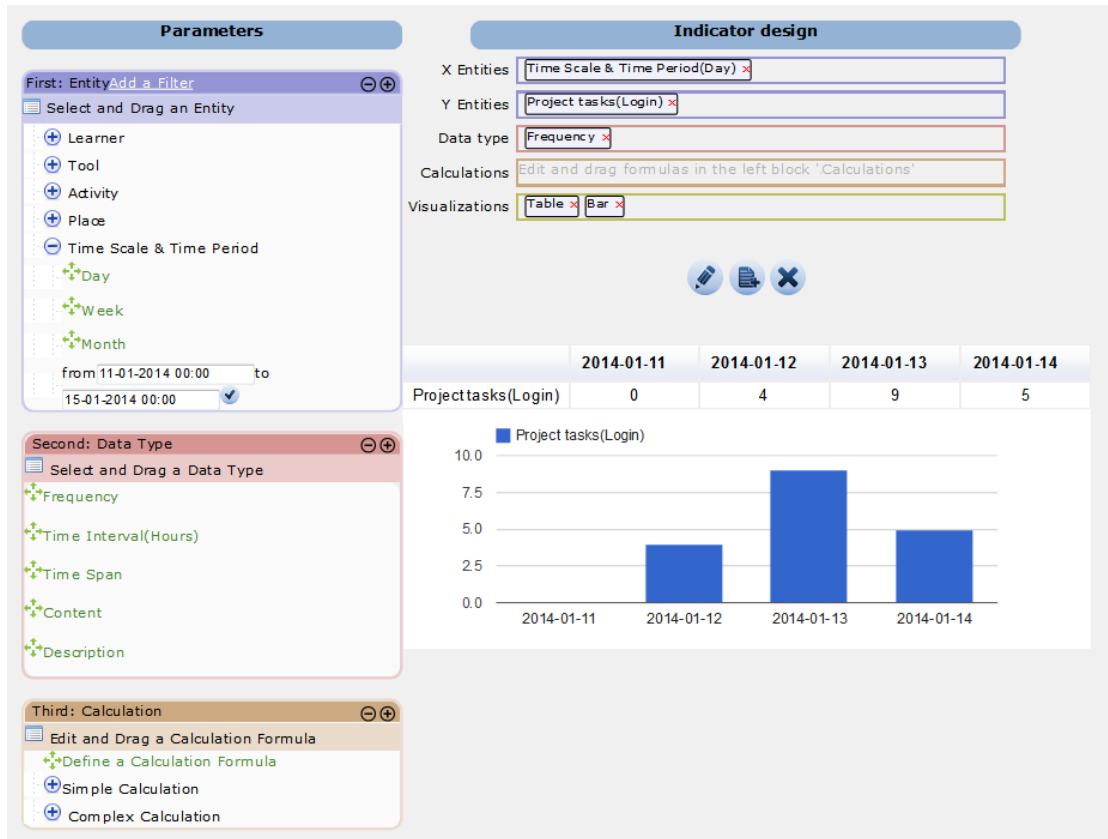


Figure 5-3 The interface to create the “students login overview” indicator (2)

DDART can create the login indicators proposed in GISMO. GISMO presents two indicators together: learners’ daily accesses and group daily login frequency. For the first indicator, each red square in GISMO represents at least one access to the system. However, the indicator created in DDART can present each login precisely and the length of the blue square specifies the duration of each login, which cannot be achieved by GISMO. The limitation is that DDART is unable to display two indicators in one diagram.

5.1.1.2 User Classification

Figure 5-4 presents a XY scattered chart. The X-Axis represents the amount of forum messages written by the user and the Y-Axis represents the amount of forum messages read by the user (Bratitsis and Dimitracopoulou 2006). The two axes are scaled from Low to High. The X-coordinate places the lowest number at the left end of the Axis (Low) and the highest number at the right end (High). The Y coordinate places the lowest number at the bottom end of the Axis (Low) and the highest number at the top end (High). The result is related to the two constituents: writing and reading forum messages. By observing this indicator, the learners may visualize their own states in the group compared with the other learners. So this indicator can help learners to discover their extreme or balanced behaviors (Arrogant: writes many messages but doesn’t read other learners’ messages. Passive: reads many messages, but does not write enough).

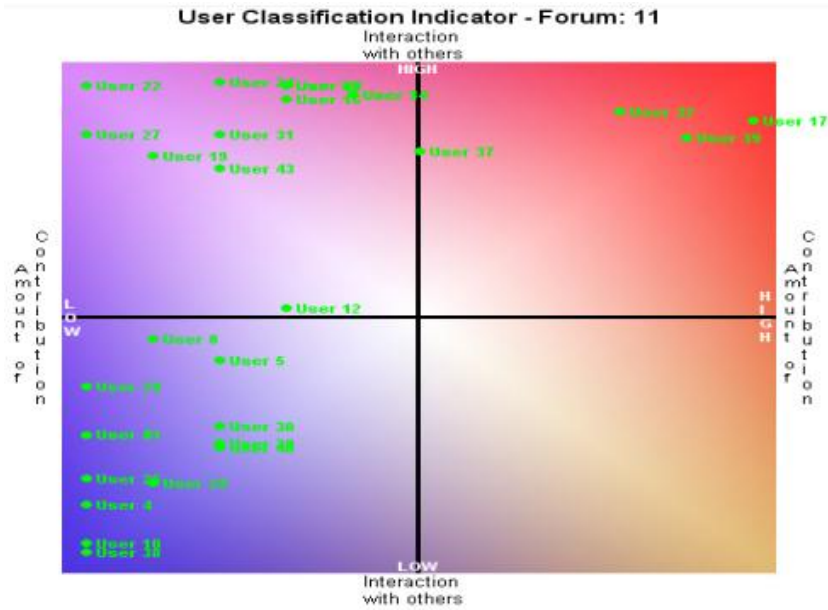


Figure 5-4 The visualization of “users classification” indicator (Bratitsis and Dimitracopoulou 2006)

Figure 5-5 is the main interface to create the “user classification” indicator with DDART. The parameters are listed in the “Indicator design” part. The entity values listed in the “X Entities” pane are “Learner (Benoit)”, “Learner (Francois)” and “Learner (Pascal)”. The entity values listed in the “Y Entities” pane are “Project tasks (WriteForumPost)” and “Project tasks (ViewForumPost)”. Each X entity values is crossed with each Y entity values separately. The “Frequency” data type allows calculating the frequency of each selected task performed by each learner. The “Description” data type can give more explanations about the indicator results. When the mouse is over the indicator results, the corresponding explanations are shown below the results in the format of text (in Figure 5-5, the text gives the detail contents of the eight forum messages written by Benoit). The indicator results are presented in the “Table” and “Bar” chart. From the Figure 5-5, we can draw the conclusions that Benoit, Francois and Pascal read more forum posts than they write.

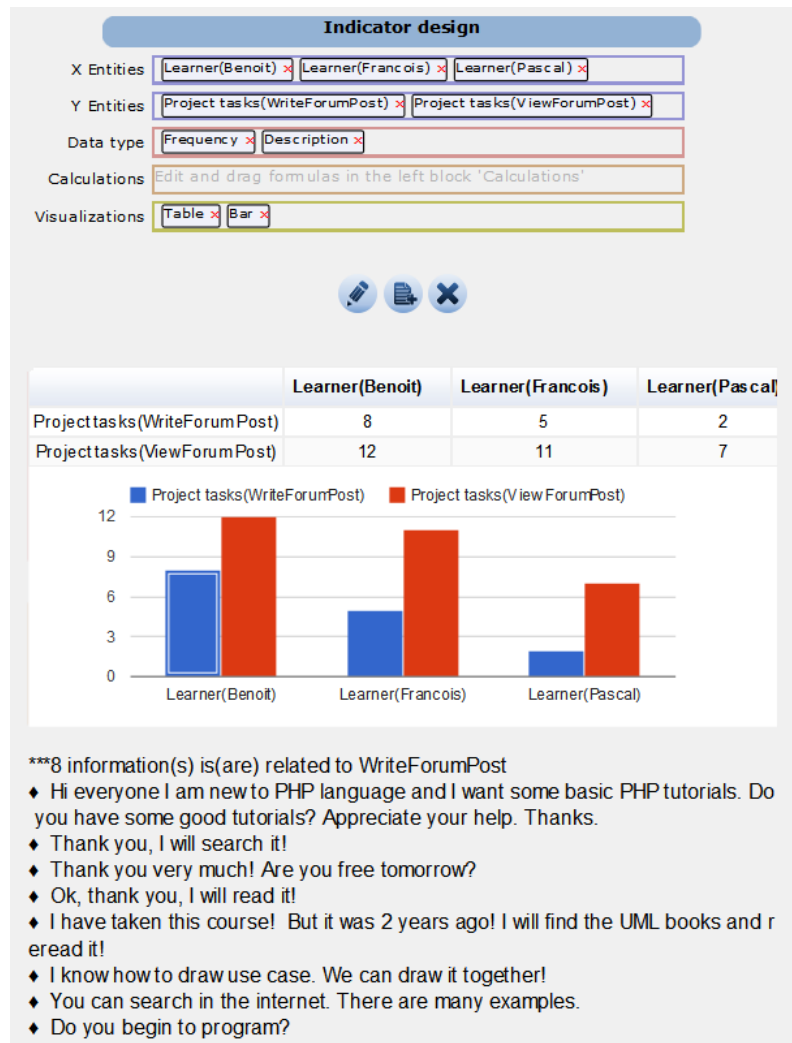


Figure 5-5 The interface to create the “user classification” indicator

In this indicator, DDART can calculate the amounts of forum posts written and read by each learner. Furthermore, DDART can explain the indicator results with detailed information, which cannot be achieved by the original indicator. However, DDART is unable to provide the same visualization mode like XY scatter chart in Figure 5-4. The scatter chart provided in DDART accepts merely the date as the X-coordinate.

5.1.1.3 Time spent on activities

Figure 5-6 is a bar chart that presents the time spent by a specific learner on six different activities. This indicator also displays the corresponding average time spent by the whole group (Santos *et al.* 2012). The X-coordinate indicates the time durations while the Y-coordinate indicates the different activities. By observing this indicator, the learner can know how his/her time is spent on the project activities and whether the time is higher or lower than the group average time. It gives hints to the learner to adjust his/her time allocation appropriately.

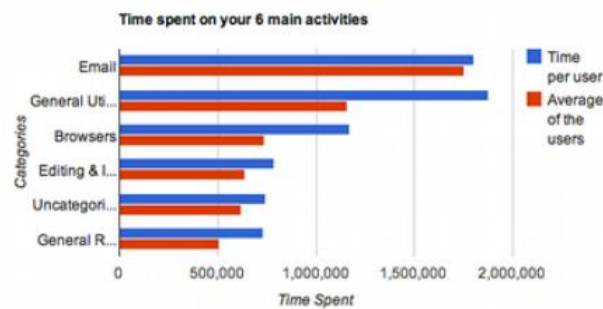


Figure 5-6 The visualization of “time spent on activities” indicator (Santos *et al.* 2012)

Figure 5-7 is the interface of DDART to create the above “time spent on activities” indicator. This indicator focuses on the “Learner” and “Activity” entities. We choose to focus on Julien and on the two types of activities: viewing wiki and programming. The “viewing wiki” activity is tracked by the system automatically while the “programming” activity is recorded in the reports manually by the learner. These two activities are listed in the “X Entities” pane and are crossed with the entity values: “Learner (Julien)” and “Learner (All Learner)” listed in the “Y Entities” pane. The data type of the indicator is “Time Interval (Hours)” so as to calculate the time spent by Julien and all the group learners on the activities of viewing wiki and programming. The indicator results are presented in the formats of “Table” and “Bar” chart. In order to calculate the group average time, it is necessary to edit a formula. The formula is edited in the “Calculation” block and is launched by dragging it to the “Calculations” pane. Therefore, in the table, a new row (named “average”) is generated to present the average time spent on each activity performed by the group. We can draw the conclusions from the Figure 5-7 that the time spent by Julien on viewing wiki and programming is less than the group average time.

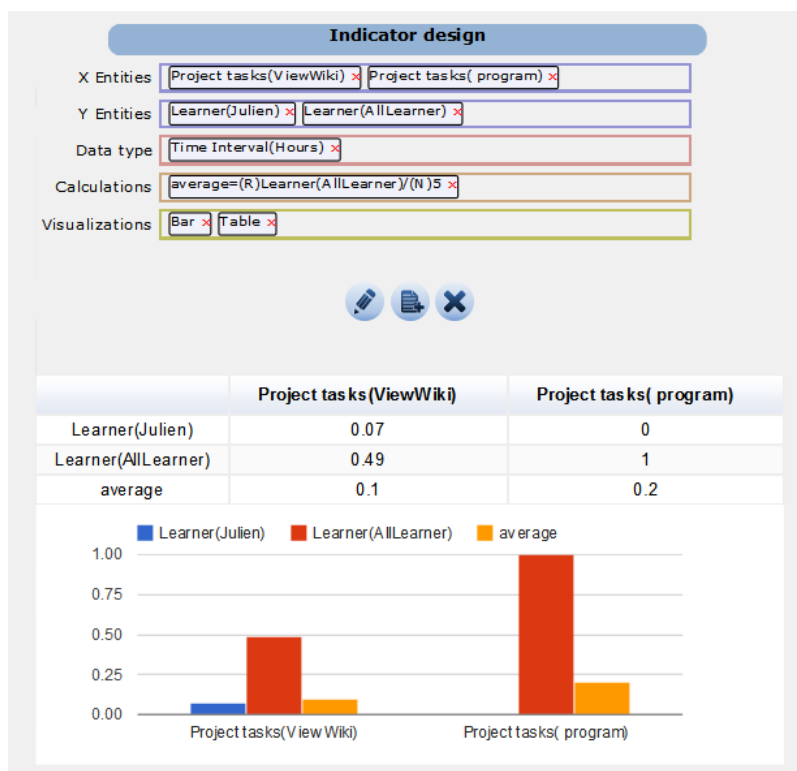


Figure 5-7 The interface to create the “time spent on activities” indicator

Compared with the original indicator, we can find that DDART can calculate the time spent by a specific learner on different activities separately. And it supplies the powerful calculation operators to compute the group average time. However, the “Learner (AllLearner)” row cannot be deleted from the “Table” and “Bar” chart visualization, because the group average value is calculated based on it. Namely, the intermediate variables cannot be removed from the visualizations in DDART.

5.1.1.4 Knowself resourceblock

Figure 5-8 presents the visualization of the “knowself resourceblock” indicator, generated by the Mirror User Profile (MUP) application⁷. The bar chart presents the total usage time spent by a specific learner on the different applications, e.g., the time spent on editing or reading Word files, the time spent on online meeting by Skype. The learner not only can compare his/her time allocation on different applications but also can discover how s/he carries out the project with the help of some special tools. In Figure 5-8, each bar represents a specific application and the length means the time spent by a specific learner.

⁷ <http://www.mirror-project.eu/mirrorsolutions/mirror-apps-status/279-mup>

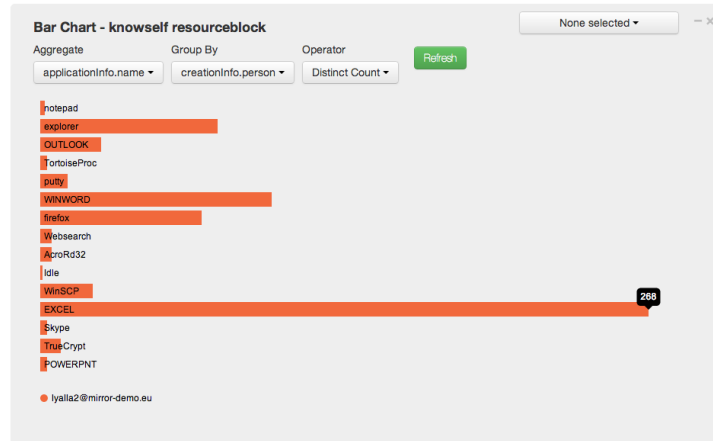


Figure 5-8 The visualization of “time spent on tools” indicator

Figure 5-9 is the interface of DDART to create the above indicator. In the “Indicator design” part, all the indicator parameters are listed. This indicator relates to “Learner” and “Tool” entities. Hence, we select several tool values and drop them in the “X Entities” pane: “Tool (Chat)”, “Tool (Forum)”, “Tool (YouTube)” and “Tool (Skype)”. We select two learners and drop them in the “Y Entities” pane: “Learner (Julien)” and “Learner (Benoit)”. The data type is “Time Interval (Hours)” to calculate the time spent by Julien and Benoit on the following applications: Chat, Forum, YouTube and Skype. The time spent on the Chat and Forum can be captured directly by the system while the time spent on YouTube and Skype is recorded manually by the learners in the reporting tool. The indicator results are visualized in the “Bar” chart. From the Figure 5-9 we can know that both Julien and Benoit spent much time on YouTube while they used rarely the Chat and Forum.

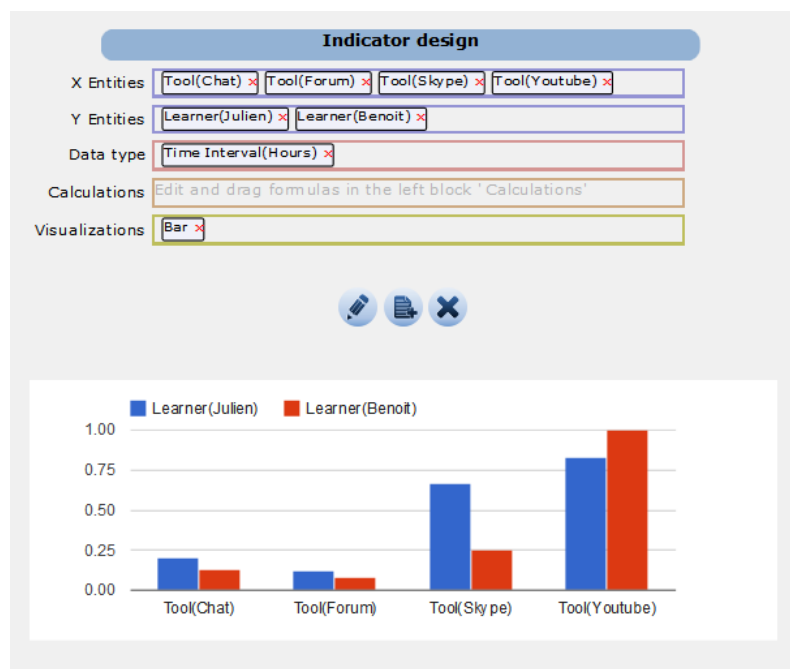


Figure 5-9 The interface to create the “knowself resourceblock” indicator

DDART enables to calculate the time spent by the learners on the different tools. Compared with the original indicator, DDART can present the results of different learners together, which cannot be achieved by the original indicator (the original indicator only can present the results related to a specific learner). It allows learner to compare his/her time allocation with the others.

5.1.2 Social indicators

5.1.2.1 Forum graph

Reffay and Chanier (2003) proposed a forum indicator (see Figure 5-10). This indicator is a directed and valued graph $G_f(A,M,P)$. 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. In the table of Figure 5-10, the corresponding number is the messages posted by the agent a and opened by agent b during a time period P in the discussion forums. In the right part of Figure 5-10, the network presents the interaction relations between learners.

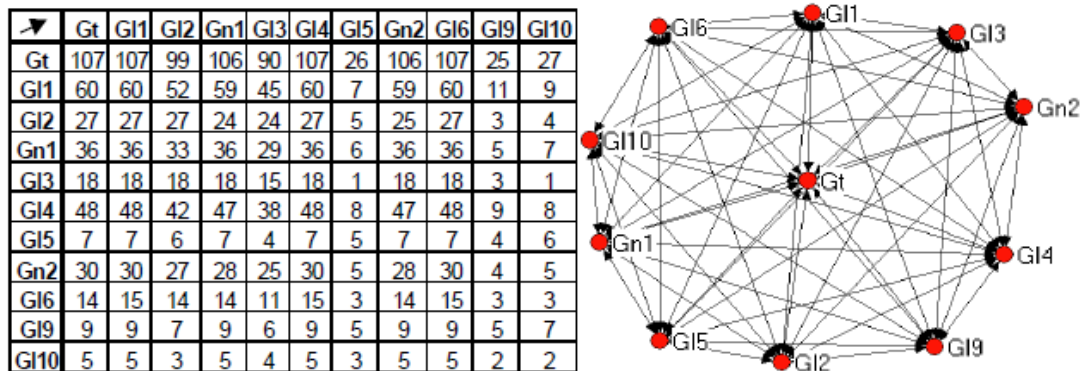


Figure 5-10 The visualization of the “social network of forum messages” indicator (Reffay and Chanier 2003)

Figure 5-11 is the interface of DDART to create the indicator described above. We suppose that this indicator presents the social connections between Julien, Sophie, Francois and Pascal. 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 visualization modes are “Social network” and “Table”. The “Frequency” data type can calculate the interaction frequencies between each learner. The “Description” data type provides the detailed interaction information when the mouse is over a specific result. For example, Julien has viewed the posts written by Francois seven times and he has replied to the posts written by Francois two times. The time period is set from 11-01-2014 to 17-01-2014 in the “Entity” block. We have to set a filter in order to focus on the interactions of viewing and writing forum posts. The indicator results are presented in the “Result” part.

From this indicator, we can know that Francois, Julien and Pascal connected frequently while nobody contacted Sophie.

	Learner(Francois)	Learner(Julien)	Learner(Sophie)	Learner(Pascal)
Learner(Francois)	8	17	0	
Learner(Julien)	9	23	0	
Learner(Sophie)	2	3	0	
Learner(Pascal)	1	10	0	

***7 information(s) is(are) related to ViewForumPost
 ***2 information(s) is(are) related to WriteForumPost
 ♦ I find a website about SQL: mysql.com
 ♦ Good, could you be responsible to draw the UML?

Figure 5-11 The interface to create the “forum graph” indicator

DDART can create “Social network” indicator to present the learners’ social states in the group. Compared with the original indicator, the indicator created by DDART can present the interaction frequencies by the thicknesses of the arrows. Furthermore, the detailed interaction information is also presented.

5.1.2.2 Peer assessment indicator

Kennedy (2005) conducted a peer assessment experiment in a group project. Figure 5-12 presents the peer assessment results of one team. If there are n members in a group, each member has a total of $100 * (n-1)$ marks and has to allocate these marks to the other member regarding to their contribution. If the score awarded for any individual exceeds 110, or is less than 90, a short justification is required. The general assessment of each learner is the average score obtained from their peers. The standard deviation is also computed. This indicator helps learners to do peer-judgments and they can know the impression they give to others.

Team 2											
Stuld	11	12	13	14	15	16	17	18	19	Total	Sdev
11		97	97	97	97	97	97	97	121	800	8.5
12	110		100	100	90	100	90	100	110	800	7.6
13	110	100		100	100	90	80	100	120	800	12.0
14	112	100	100		95	95	90	100	108	800	7.2
15	100	100	100	100		100	100	100	100	800	0.0
16	120	95	95	100	100		95	95	100	800	8.5
17	100	100	100	100	100	100		100	100	800	0.0
18	115	100	100	100	95	95	90		105	800	7.6
19	125.6	96.5	96.5	96.5	91.5	96.5	100.5	96.5		800	10.6
Total	892.6	788.5	788.5	793.5	768.5	773.5	742.5	788.5	864	7200	
Avg	111.6	98.6	98.6	99.2	96.1	96.7	92.8	98.6	108.0		
Sdev	8.9	2.1	2.1	1.5	3.9	3.5	6.8	2.1	8.6		

Figure 5-12 The visualization of “peer assessment” indicator (Kennedy 2005)

Figure 5-13 shows the interface to create the above “peer assessment” indicator in DDART. In this case, we focus on the assessments between three learners: Benoit, Francois and Julien. Hence, in the “Y Entities” pane, three participants are listed: “Learner(Benoit)”, “Learner (Francois)” and “Learner (Julien)”. In the “X Entities” pane, the judgment actions “Judge(Benoit)”, “Judge(Francois)” and “Judge(Julien)” are selected. All the assessments are recorded manually in the learners’ reports. Learners can assess each other group member by selecting a value from a five likert scales (from “very good” to “not good at all”) which correspond to the numbers from 2 to -2. The data type is the “Content” of the assessments which are written in learners’ reports. The visualization mode is “Table”. In order to calculate the average assessments, two formulas are edited to calculate the total scores (row “SumC” in the table visualization) and the average scores (row “AVG” in the table visualization) obtained by each learner. The “Description” data type presents the justifications of the judgments below the indicator results. For example, Francois thinks Benoit is good (corresponding number is 1) “*because he is active*”. From this indicator, we can know that Francois get the lowest average assessment (= -1) from Benoit and Julien while the average assessments obtained by Benoit and Julien are almost the same (1 and 1.5 respectively).

The interface for creating a 'peer assessment' indicator is shown. It consists of two main panels: 'Parameters' and 'Indicator design'.

Parameters Panel:

- First: Entity Add a Filter:** A tree view showing entities: Learner, Tool, Activity, Place, Time Scale & Time Period (with sub-items Day, Week, Month). A date range is set from 12-01-2014 00:00 to 13-01-2014 00:00.
- Second: Data Type:** A list of data types: Frequency, Time Interval(Hours), Time Span, Content, Description.
- Third: Calculation:** Options for Simple Calculation and Complex Calculation.

Indicator design Panel:

- X Entities:** Judge(Benoit), Judge(Francois), Judge(Julien)
- Y Entities:** Learner(Benoit), Learner(Francois), Learner(Julien)
- Data type:** Content, Description
- Calculations:** SumC=sumup column, AVG=(R)SumC/(N)2
- Visualizations:** Table

Table:

	Judge(Benoit)	Judge(Francois)	Judge(Julien)
Learner(Benoit)		-1	2
Learner(Francois)	1		1
Learner(Julien)	1	-1	
SumC	2	-2	3
AVG	1	-1	1.5

Text below table:

***1 information(s) is(are) related to Benoit
♦ I judge that Benoit is good because he is active.

Figure 5-13 The interface to create the “peer assessment” indicator

Compared with the original indicator, the standard deviation cannot be computed in DDART because it does not provide the calculation functions of extracting the square root and power. Furthermore, when learners judge their peers in the reporting tool, the judgment results are not awarded. Namely, they only can select one judgment value from the range of the five likert scales and the rest value of the judgment cannot be awarded to others.

5.1.3 Cognitive indicators

Michel, Lavoué and Pietrac (2012) proposed an indicator which can present the current levels of the knowledge that learners have acquired, in comparison with their target levels defined in the project plans. For example, in Figure 5-14, one learner indicates that his current level of “Database” is in the second level and his target is the fourth level. Namely, he needs to progress in “Database” for two more levels. This indicator encourages learners to keep the motivations in the projects and support them to know their own cognitive progresses.

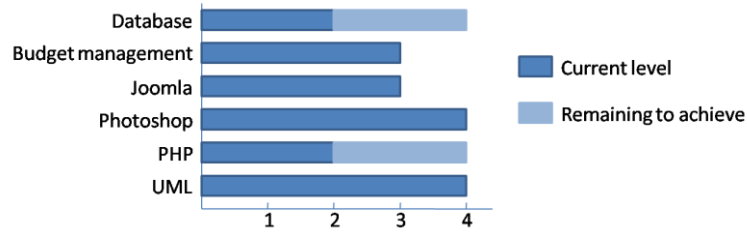


Figure 5-14 The visualization of “knowledge development” indicator (Michel *et al.* 2012)

Figure 5-15 presents the interface to create the “knowledge development” indicator in DDART. In the reporting tool, learners can set their target levels of knowledge/skills in their plan reports. With the development of the project, they can assess and record their knowledge/skill levels in the activity reports at any time. The reporting tool provides a five likert scales (from “very good” to “not at all good”, correspond 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 12-01-2014 to 15-01-2014 in the “Entity” block. The target plan of documentation and MySQL as well as the corresponding judgments are placed in the “Y Entities” pane. A filter is necessary to focus on the information about Benoit. The data type is the “Content” because the results come from the learners’ report contents. The visualization modes are “Bar” chart and “Table”. The data type “Description” 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.

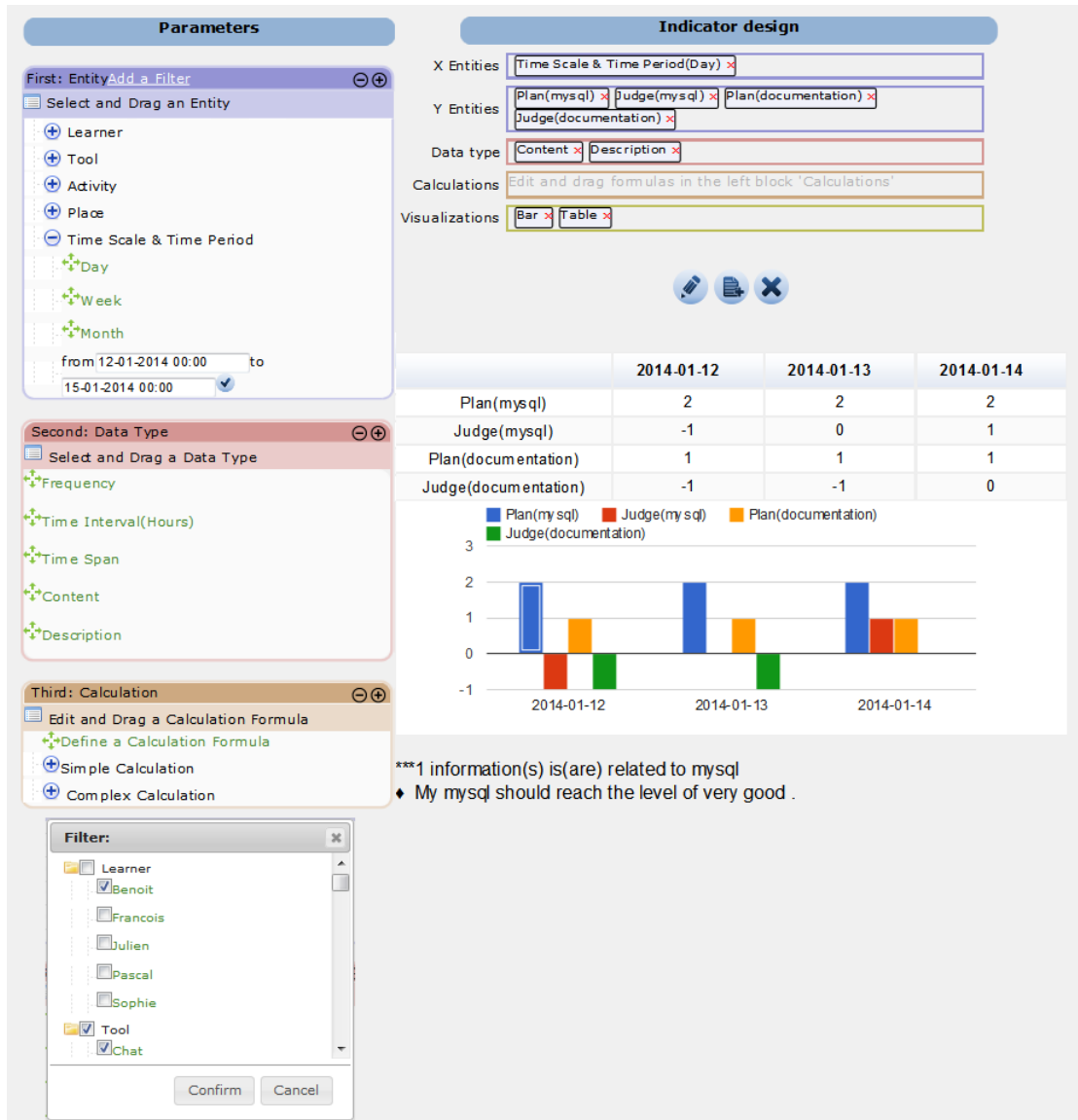


Figure 5-15 The interface to create the “knowledge development” indicator

Compared with the original indicator, DDART enables to present the learners' goals of knowledge/skills and their actual progresses. It supports learners to follow their cognitive progresses based on the date. However, DDART is unable to present the goal and the actual progress in one bar. Hence, it is a little difficult to visualize the gaps between the goals and the current progresses at one glance.

5.1.4 Affective indicators

Ferguson, Shum, and Crick (2011) proposed an indicator (Figure 5-16) which can reflect learners' affective states during the learning. Learners can choose one of five emoticons to reflect their emotional states, ranging from “going great” to “it's a disaster”. When they choose an emotional state, they have to write the

explanations of their choices. All the information is recorded as a blog post. Learners can observe their mood trends over time in the line chart.



Figure 5-16 The visualization of “mood view” indicator (Ferguson *et al.* 2011)

Figure 5-17 presents the interface of DDART to create the above “mood view” indicator. Learners can assess their moods in their reports by selecting a state from a five likert scales (from “very good” to “not good at all”) which correspond to the numbers from 2 to -2. We suppose this indicator aims to observe the mood trend of Benoit from 12-01-2014 to 17-01-2014. In the “X Entities” pane, the “Time Scale & Time Period (Day)” entity value is listed so that we can observe the mood trend over time. The time period is set from 12-01-2014 to 17-01-2014 in the “Entity” block. In the “Y Entities” pane, we pay attention to the action “Judge (mood)”. We have to set a filter in order to focus on the moods of Benoit. The data type is the “Content” because the information on moods is written by Benoit in the reporting tools. The “Description” data type offers detailed information about Benoit’s moods. The mood trend is presented in a line chart. From this indicator, we can know Benoit was sad in 13-01-2014 “*because he is tired*”.

The screenshot shows the DDART interface for creating a 'mood view' indicator. The interface is divided into two main sections: 'Parameters' and 'Indicator design'.

Parameters Section:

- First: Entity:** A panel titled 'Add a Filter' with a 'Select and Drag an Entity' button. It lists entities: Learner, Tool, Activity, Place, Time Scale & Time Period (with sub-items Day, Week, Month), and a date range from 12-01-2014 00:00 to 17-01-2014 00:00.
- Second: Data Type:** A panel titled 'Select and Drag a Data Type' with a 'Select and Drag a Data Type' button. It lists data types: Frequency, Time Interval(Hours), Time Span, Content, and Description.
- Third: Calculation:** A panel titled 'Edit and Drag a Calculation Formula' with a 'Define a Calculation Formula' button and options for Simple Calculation and Complex Calculation.
- Filter:** A separate window showing a tree view of entities: Learner (with sub-items Benoit, Francois, Julien, Pascal, Sophie), Tool (with sub-items Chat), and buttons for Confirm and Cancel.

Indicator design Section:

- X Entities:** Time Scale & Time Period(Day)
- Y Entities:** Judge(mood)
- Data type:** Content, Description
- Calculations:** Edit and drag formulas in the left block 'Calculations'
- Visualizations:** Line

Below the design fields, there are three icons (edit, print, close). A line graph titled 'Judge(mood)' shows data points for dates from 2014-01-12 to 2014-01-16. The y-axis ranges from -1 to 3. The data points are approximately: (2014-01-12, 2), (2014-01-13, -1), (2014-01-14, 1), (2014-01-15, 2), (2014-01-16, 2). Below the graph, there is a text box with a redacted name and a learner's self-assessment: 'I self assess that my mood is not very good because i am so tired today .'.

Figure 5-17 The interface to create the “mood view” indicator

Like the original indicator, DDART enables learners to record their moods and to create indicators to present the mood trends. Learners' explanations are provided to give more information. However, DDART cannot provide learners with the emoticons to select, which is more meaningful than the five likert scales.

5.1.5 Discussion

From the above illustrations of use of DDART, we can find that DDART supplies the features for learners to create customizable indicators related to activity, cognition, emotion and social categories. It gives learners the possibility to calculate complex indicator results by allowing editing formulas. It also enables to present more explanations about the indicator results. However, we

highlighted some limitations of DDART. It is unable to display two indicators in one diagram. The “Scatter” chart supplied in DDART accepts merely the date as the X-coordinate. Some complex calculation operators (e.g. power, extract a root, standard deviation and so on) are missing in DDART. Furthermore the intermediate variables cannot be removed from the indicator visualizations.

Even though DDART has some limitations and can be improved further, we think DDART is valid regarding to its possibility to create several kinds of existing indicators. We also wanted to carry out an experiment with real users to test the usability and utility of DDART. In the next section, we present the experiment and its results.

5.2 Experiment

We have conducted an experiment with several learners in order to test the usability and utility of DDART. The experiment is semi-controlled and the participants have to follow a scenario to complete different tasks with DDART. Considering that the experiment is semi-controlled and the experiment time is short, we are aware that we only test the perceived utility. The traces left by the participants in DDART and the answers of the survey are analyzed and discussed. At last, we draw the conclusions and propose some improvements of DDART.

5.2.1 Experiment research questions

The object of this experiment is to evaluate the usability and the utility of DDART. Krug (2000) provided a definition that “*Usability really just means making sure that something works well: that a person of average (or even below average) ability and experience can use the thing—whether it’s a website, a fighter jet, or a revolving door—for its intended purpose without getting hopelessly frustrated.*” Generally speaking, usability “*is usually considered the ability of the user to use the thing to carry out a task successfully*” (Albert and Tullis 2013), which includes the aspects of ease to learn, efficient to use, easy to remember, lack of errors, and subjectively pleasing (Nielsen 1994). Utility is a measure to test whether the system functionalities can help the learners to reach their goals, which refers to the satisfaction of learners’ needs. In order to be more targeted, we decompose the usability and utility into several precise research questions as follow:

- **Usability:**

1. ***Is DDART easy to use?***

Ease of use is one of the main measures of usability. A system which can be handled easily will reduce the work load of learners and increase learners’ motivations to use it. We want to know whether DDART is easy for the participants to use. If they think it is hard to use, it is important to identify precisely the difficulties.

2. ***Is DDART easy to learn?***

Learnability is the extent to which something can be learned (Albert and Tullis 2013). When learners use most of new computer tools, it is necessary to learn how to handle them. According to their different complexities, the periods of learning are different. We want to know whether the participants can learn how to use DDART easily.

3. ***Are the indicators created in DDART readable?***

The indicators created in DDART are presented into diagrams. Do participants can understand these diagrams? Do they can extract the important information from the indicators?

● ***Utility:***

1. ***Which indicator is useful?***

DDART allows users to create plenty of customizable indicators. From the participants' points of views, which indicators are the most useful in PBL?

2. ***Is DDART useful in supporting reflexivity and managing the project?***

DDART enables learners to reflect and monitor the projects by creating customizable indicators. Is it helpful to manage projects by observing the indicators?

5.2.2 Participants

Thirteen participants (5 women and 8 men) in the Institut National des Sciences Appliquées de Lyon (INSA Lyon) ranging in age from 21 to 28 years old, voluntarily participated in this experiment. All of them were assigned to the same experimental scenario and they never used DDART before. During the experiment, one participant gave up the experiment halfway. Therefore, the data of the rest twelve participants are taken into account.

According to the different professions, we classify them into two groups:

The ***engineering student group*** is composed of seven engineering students. They are coming from the department of Industrial Engineering of INSA Lyon. All of them have joined a project management course in which they have been required to carry out a project according to the PBL processes with the help of a dashboard.

The ***research student group*** is composed of five research students from the SICAL research team of the LIRIS laboratory. In fact, research students are actually PhD students or Master students. They have learnt some PBL research skills because they work in project context with one or more supervisors or other researchers.

5.2.3 Materials

We placed the participants in a specific context. They belong to a PBL group that has carried out a project for a week. We have simulated the traces of the learners'

activities in the PBLMS, for example, by posting messages in the Chat tool, writing document in the Wiki tool, reading the forum messages in Forum tool, recording the activities in the reporting tool, creating indicators in the dashboard. All the participants do the experiment based on these fundamental data.

The participants are required to use a computer installed with a browser to access to the Internet. An online scenario (Annex E) is supplied to all the participants to guide the experiment. After the experiment, an online survey composed of the "SUS" (The System Usability Scale) questionnaire (Brooke 1996) and several open questions are supplied to the participants (Annex F). The feedbacks from the survey and the participants' traces left in the system are analyzed and discussed.

5.2.4 Procedure

Before the experiment, an online scenario is supplied to the participants. The scenario (see Table 5-1) is composed of three parts: context description, tutorial videos and the tasks to carry out.

Table 5-1 The scenario of the experiment

Scenario		Description
1. Context description		This part gives the information about the project context and the group members. The objective of the project is to develop a small website. This group has five members: Sophie, Julien, Pascal, Benoit and François. The project begins from 11, Jan 2014 to 16, Jan 2014.
2. Tutorial videos		Two tutorial videos are uploaded in the YouTube ⁸ to teach the participants how to use the reporting tool and the dashboard. Each video lasts for 10 minutes.
3. The tasks	Observe the predefined indicators	Several indicators are created in advance in the dashboard. Learners are required to observe these indicators and to answer the questions related to the indicators, such as who is the most active in the group, how long is your weekly average work time.
	Write a report	Each participant is required to write a report in the reporting tool. Then they need to read other members' reports and write some comments on the reports.
	Create two indicators	Participants are required to create two new indicators in DDART according to some requirements given in the form of text. An additional diagram is supplied for the first indicator while the second indicator has no diagram.

All the participants had to do the experiment by following the above scenario. The participants in the engineering student group followed the experiment at a distance; hence no help was given to them during the experiment.

⁸ Reporting tool: <https://www.youtube.com/watch?v=PC9OwFoDbAA>

Dashboard: https://www.youtube.com/watch?v=_yIYlyc9DIw

The participants of the research student group followed the experiment in the laboratory. A technician supplied a technical support when they encountered problems in the task of creating indicators.

During the experiment, the system recorded the experimental traces automatically. These traces were used for the data analysis. Table 5-2 presents the attributes of the experimental traces.

Table 5-2 The attributes of experimental traces

Attributes	Description	
Trace id	The id of the trace.	
User Id	The id of the participant who produces this trace.	
Time	The time of producing this trace.	
Action	Accomplished in the reporting tool	Write the report title, save report, add/delete a sentence model, begin/end to view others report, begin/end to write comments
	Accomplished in the Dashboard	Begin to create indicator, save indicator, drag (delete) X entity values/Y entity values/Visualization modes/Data type/Formulas, set filter.
Value	The values of the actions, such as the comments written for the reports, the id of semi-structured sentence models selected by the participants, the values of the X entity values/Y entity values/Visualization modes/Data type/Formulas dragged by the participants, the value of the filter.	

After the participants accomplished the experiment tasks, they needed to complete the survey which is composed of two parts: ten questions coming from the SUS to test the usability of DDART and some open questions which allowed learners to express their views on the system in order to test its utility.

5.2.5 Results analysis

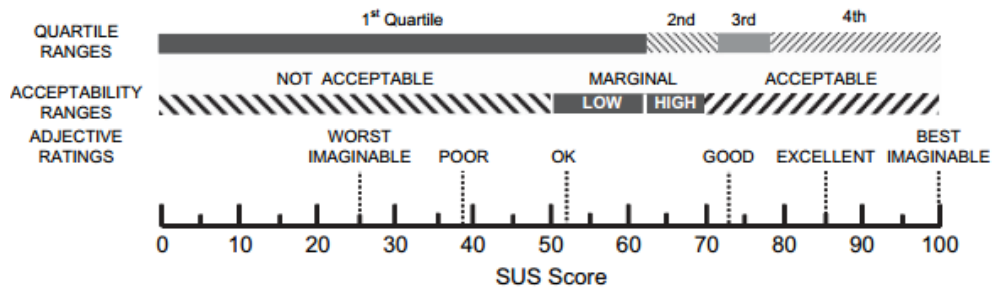
5.2.5.1 Usability

We calculated the average scores of SUS of each group and the average for the two groups (see Table 5-3). We observe that the SUS scores of the two groups are almost the same.

Table 5-3 The average scores of SUS

Engineer student group	Research student group	Average
53.92857	54.5	54.16667

In fact, Bangor, Kortum, and Miller (2008) proposed a set of acceptability ranges according to the scores of SUS, which is divided into “not acceptable”, “marginal” and “acceptable” (see Figure 5-18). The SUS score of DDART is in the range of “marginal” (that means between “ok” and “good”).

Figure 5-18 A comparison of SUS scores (Bangor *et al.* 2008)

Regarding the research questions, we set three criteria in order to analyze the reasons why DDART is “marginal”: ease of use, learnability and readability.

Ease of use of the reporting tool

The ease of use is an attribute to indicate how easy or how difficult is to use a tool to carry out some tasks. We mainly use the experimental traces to analyze the ease of use of the reporting tool (see Table 5-4). We calculate five indexes as follows:

1. **The success ratio of writing the report (SR_R)**. The success means participants can write a report and save it successfully in the reporting tool. It is calculated by the following equation:

$$SR_R = \frac{M}{N}$$

Where: N is the group size; M is the number of participants who write the report and save it successfully. The parameter M is calculated depending on the reporting traces. If there is a new report stored in the XML database by a participant, we interpret that this participant achieved to write a report successfully.

2. **The average time of writing the report (AT_R)**. The beginning time of writing a report corresponds to writing the title of the report and the end time corresponds to saving the report. It is calculated by the following formula:

$$AT_R = \frac{\sum_{i=1}^M (ET_i^R - BT_i^R)}{M}$$

Where: M is the number of participants who write the report and save it successfully, ET_i^R is the end time of writing a report by participant i , BT_i^R is the beginning time of writing a report by participant i . The beginning time and end time are extracted from the “time” attribute of the experiment traces, in which the “action” attribute is “write the report title” and “save report”.

3. **The average number of semi-structured sentences in the report (AS_R).** It is calculated by the following formula:

$$AS_R = \frac{\sum_{i=1}^M S_i}{M}$$

Where: M is the number of participants who write the report and save it successfully, S_i is the quantity of sentences written in the report by participant i . S_i is calculated from the experiment traces. For the report written by participant i , S_i can be computed by subtracting the amount of the traces in which the “action” attribute is “delete sentence model” from the amount of the traces in which the “action” attribute is “add sentence model”.

4. **The average time to write a sentence (AT_S) in the report.** It is calculated by the following formula:

$$AT_S = \frac{AT_R}{AS_R}$$

5. **The average valid operations ratio (VR_R).** The valid operations are the right actions to write/comment report, for example, write the report title, add/delete a sentence model, save the report, view others report, write comments. The invalid operations include filling the wrong contents into the sentence models, using the wrong format to fill in the sentence models. It is calculated by the following formula:

$$VR_R = \frac{\sum_{i=1}^M (V_i^R / T_i^R)}{M}$$

Where: M is the number of participants who write the report and save it successfully, V_i^R is the amount of valid operations of writing the report by participant i , T_i^R is the amount of all operations of writing the report by participant i . V_i^R and T_i^R are calculated based on the attribute “action” of the experimental traces and the reporting traces.

Table 5-4 The ease of use of the reporting tool

Index	Engineering student group	Research student group
Success ratio of writing report	100%	100%
Average time of writing report	5'30''	4'01''
Average number of sentences in the report	3 sentences	2 sentences

Average time of each sentence	1'50''	2'
Average valid operations ratio	95.24%	95%

From the Table 5-4, we observe that:

1. ***The functionalities of the reporting tool are easy to use.*** The success ratios of writing the report of the both groups are 100%. All the participants in two groups success to record their activities realized out of the PBLMS, their assessments, and reflections by using the reporting tool. The valid operations ratios of the two groups are up to 95% and prove that the features of the reporting tool are easy to handle with. However, some participants feel constrained during the reporting. A participant thought (translated from French): *“I feel constrained when I input the information of activities and judgments”*. Another participant said (translated from French): *“It is a risk if we change the name of a specific skill and the interests of following its progresses will loss.”*
2. ***The time of writing a report is long.*** The average time of writing a report of engineering student group (5'30'') is higher than the research student group (4'01''). But the participants of the engineering student group write one more sentence (3 sentences) than the research student group (2 sentences) averagely. The average time of writing one sentence of the two groups is close to two minutes. We think that two minutes is long to record one piece of information, especially when learners have much information to record at once. Some participants also mentioned this point in the open questions of the survey. One participant said: *“This is a powerful system but it takes too much time to enter the data for a project.”* Another participant wrote (translated from French): *“I spend much time to input information in the system which is not productive.”*
3. ***The semi-structured sentence models are understandable.*** The valid operations ratios of the two groups are up to 95%. By observing the invalid operations, we find that two participants wrote unsuitable information into two semi-structured sentences while the other participants wrote appropriately. Therefore, we think that the semi-structured sentence models are understandable by most of the participants.

From the above discussion, we can find that the main limitation of the reporting tool is the input method. The participants have to spend a long time to select the semi-structured sentence models and to fill in these models based on their own states manually, which is a constraint of the flexibility of the reporting tool. The input method restrains learners to input more information in a short time. However, the semi-structured sentences have the advantages in supporting reflection and collecting the information used for calculating indicators. Hence,

in the future research, it is necessary to improve the method to input the information.

Ease of use of the dashboard

In order to evaluate the ease of use of the dashboard, we analyzed the experimental traces related to the creations of two customizable indicators and obtained the results presented in Table 5-5. For each indicator, we mainly focus on five indexes:

1. **Success ratio of creating the indicator (SR_I)**. The “success” means that the participants can create the specified indicator successfully. It is calculated by the following formula:

$$SR_I = \frac{C}{N}$$

Where: N is the group size; C is the number of participants who can create the indicator successfully. The parameter C is calculated based on the experimental traces. If the experimental traces show that the participant selected the right indicator parameters (entity values, data type, visualization mode, filter, mathematic formula) and saved the indicator in the database, we estimate that this participant achieved to create this indicator successfully.

2. **Average time of creating an indicator (AT_I)**. The beginning time of creating an indicator is the time when the participants enter into the corresponding interface. The end time is the time when the participants save the indicator. It is calculated by the following formula:

$$AT_I = \frac{\sum_{i=1}^N (ET_i^I - BT_i^I)}{N}$$

Where: N is the group size, ET_i^I is the end time of creating an indicator by participant i , BT_i^I is the beginning time of creating an indicator by participant i . for each indicator, ET_i^I and BT_i^I are extracted from the “time” attribute of the experimental traces, in which the “action” attribute is “begin to create indicator” and “save indicator”.

3. **Average valid operations ratio of creating indicator (VR_I)**. The valid operations are selecting the correct indicator parameters and deleting the wrong parameters. The invalid operations are selecting the wrong indicator parameters and deleting the right parameters. It is calculated by the following formula:

$$VR_I = \frac{\sum_{i=1}^N (V_i^I / T_i^I)}{N}$$

Where: N is the group size, V_i^I is the amount of valid operations of creating a specific indicator by participant i , T_i^I is the amount of all operations of creating the indicator by participant i . V_i^I and T_i^I are calculated from the experiment traces, in which the “action” attribute is related to drag or delete X entity values/Y entity values/Visualization mode/Data type/Formulas and set filter.

4. **Average invalid operation ratio of each parameter (IO).** This average is calculated by the following formula:

$$IO = \frac{\sum_{i=1}^N (IP_i/TP_i)}{N}$$

Where: N is the group size, IP_i is the amount of invalid operations related to a specific indicator parameter performed by participant i , TP_i is the amount of all the operations related to this specific indicator parameter performed by participant i . TP_i and IP_i are calculated based on experiment traces.

5. **Efficiency (E).** Efficiency 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. It is calculated by the following formula:

$$E = \frac{SR_I}{AT_I}$$

Table 5-5 The ease of use of the dashboard

Index		Engineering student group			Research student group		
The first indicator	Success ratio	57%			100%		
	Average time	12'31''			7'39''		
	Efficiency	4.56			13.07		
	Average valid operations	70.7%	Average invalid operations of "X/Y entities"	27.9%	83.71%	Average invalid operations of "X/Y entities"	16%
	Average invalid operations of "Data type"		35%	Average invalid operations of "Data type"		13.3%	
	Average invalid operations of "Visualizations"		11.9%	Average invalid operations of "Visualizations"		10%	
The second indicator	Success ratio	83%			100%		
	Average time	6'55''			5'04''		
	Efficiency	11.99			19.76		
	Average valid operations	74.27%	Average invalid operations of "X/Y entities"	27.7%	89.28%	Average invalid operations of "X/Y entities"	11.1%
	Average invalid operations of "Data type"		5.56%	Average invalid operations of "Data type"		0%	
	Average invalid operations of "Visualizations"		0%	Average invalid operations of "Visualizations"		0%	

From Table 5-5, we can conclude that:

1. ***The dashboard is not easy to use for the novices.*** By observing the indexes of success ratio, average time and efficiency of the engineering student group, we can find that these three indexes are not positive. We think that the functions of the dashboard to create the indicators are not easy to be handled by the participants. It is difficult for the participants to understand the objectives of each function and how each parameter works. The time is another objective constraint. Because the experiment time is short, it is a challenge for the participants to learn how to use the dashboard and to create a specified indicator by themselves. Therefore, we can confirm that it is hard for the novices to create customizable indicator by using the dashboard at the beginning. This point also can be proved by the feedbacks of the survey. One participant wrote (translated from French): *“I think the system is powerful but it is a little complex to create indicators.”*

Comparing the two groups, the research student group spent less time on the indicator creation than the engineering student group. The success ratio of indicator creation and the efficiency value of the research student group are higher than the engineering student group. We think that the main reason is because the research student group has a technical support. Hence, we can find that it is necessary to supply learners with sufficient tutoring or trainings before they use DDART to create indicators.

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. We analyzed the experiment traces further in order to explain this result. It is necessary to set a filter to create the first indicator and we observe that all of the participants who failed in creating this indicator did not set the filter. For the second indicator, it was not required to set the filter. Hence, we think that the main reason might be that the participants did not understand the purpose of the filter so that they were not aware of the necessity to set the filter in some cases. We also think that the filter is not well designed and it has to be improved so as to be easier to find and understand.

2. ***The invalid operations are highly related to the setting of the “X/Y entities” and “Data type” parameters.*** When both groups create the first indicator, the invalid operation ratios of setting the “X/Y entities” and “Data type” parameters are high. However, when both groups create the second indicator, the invalid operation ratios of setting parameter “Data type” are significantly decreased and the invalid operation ratios of setting parameter “X/Y entities” are almost as the same as the first indicator. We think there might be two reasons:

- 1) Some terms of the “Data type” are ambiguous so that the participants were confused. The participants could not understand the meanings of some words of the “Data type”. Some of them are understood literally while some

words are ambiguous, for example, “content” and “description”, “time interval” and “time span”. This can be proved from the survey of one participant, he wrote (translated from French): “*Some words are not understandable (for example: content vs description). But after a few of practices, some problems can be eliminated.*”

2) The participants did not do the project in real life. All the experiment data, such as writing wiki, chatting and organizing the meeting, are simulated and were put into the system before the experiment. The participants are not familiar with the entity values. Hence, the invalid operation ratios of setting the “X/Y entities” parameter of the two indicators did not change too much.

By analyzing the feedbacks of the survey, we can also highlight some points of views related to the usability of DDART.

- 1 ***The interface design needs to be improved.*** Some participants disclosed some inappropriate designs in the interface. One participant said (translated from French): “*It is not very convenient to drag and drop and the panes should be higher. The filters can be improved with a general box to check or uncheck all the subsets. The reports are not very intuitive.*” And another participant wrote (translated from French): “*After all, the dashboard is easy to construct. But regarding to the interface, it is not very intuitive because we need to scroll to see the lower part of the indicator.*” Hence, it is necessary to focus on the user experience improvement in the future research.
- 2 ***The time of loading the web pages is long.*** We find some participants complain in the survey that it is slow when the system load the page of the individual dashboard, such as (translated from French) “*The system is slow display.*”; “*It takes long time to load the pages.*”; “*Slow Display is not friendly.*” In fact, in the background, we need to do plenty of “select” and “insert” SQL queries into the database in order to integrate the activity traces and reporting traces and to calculate the indicators. Therefore, a solution is to host the system in a virtual machine which has a faster bandwidth. Another solution is to display some animations (e.g. a countdown clock) when the page is loading.

From the above discussion, we highlight the fact that the dashboard enables learners to create customizable indicators without coding. However, it is not easy for the novices to create customizable indicators in the dashboard. By analyzing the experiment traces, we can find that one of the limitations is the filter function. Most of the participants did not understand the aim and the operations of the filter. In the future research, we plan to redesign the filter function or to supply more online helps. Another limitation is related to the interface design. Some terms are ambiguous so that some participants cannot understand them. We plan to replace these terms with more understandable terms with the help of end users. The interfaces also need to be improved in order to strengthen the user experience. The third limitation is related to the experiment

design. Considering that the experiment time was limited, participants used the dashboard in a simulated context. Hence, it was hard for them to understand the experiment context in a short time. At last, it is slowly to load the dashboard application. Because there are a lot of calculations in the background, we can equip with a faster bandwidth or display some animations when the page is loading.

Learnability

Learnability assesses how much time or effort is required to achieve maximum efficiency (Albert and Tullis 2013). Regarding to the feedbacks in the survey, we extract two questions from the SUS (Question 4: “*I think that I would need the support of a technical person to be able to use this system.*” and Question 10: “*I needed to learn a lot of things before I could get going with this system.*”), which are related to the learnability, to study the participants' opinions.

Figure 5-19 presents the opinions of the engineering student group. From this figure, we observe that three of the engineering students think they need technical support to use the system. However, five students think they do not need to learn a lot before using the system. It proves the fact that the participants can master the system quickly if some helps are provided at the beginning. In fact, the system is hard to handle at first so that the participants need help. But it does not require users to learn much. When users are familiar with the system, they can do better to create the indicators. This point also can be proved by the answers of the survey. One participant mentioned in his survey: “*With a short teaching and some practices, this system is easy to handle, and can be powerful.*” And another one said: “*The system made users feel confused and complicated at the beginning. But after using a while and being familiar with it, it's clearer and easier to use.*”

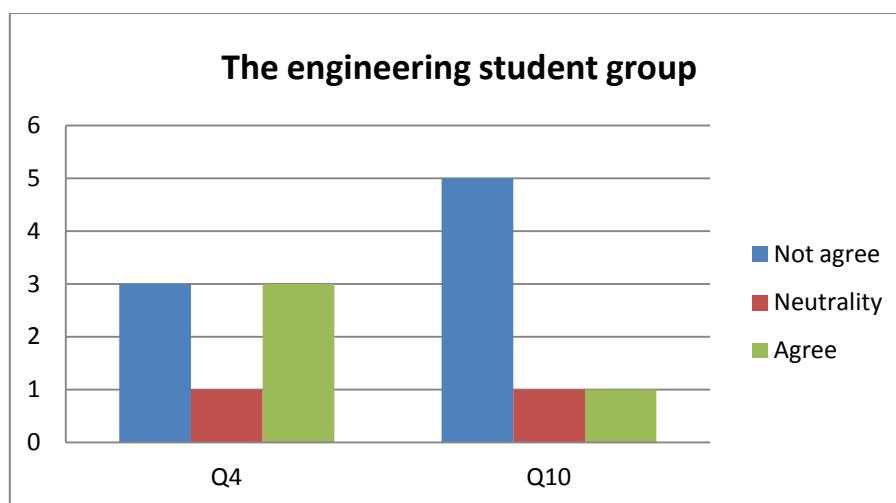


Figure 5-19 The opinions about learnability of the engineering student group

As illustrated in the Figure 5-20, all the research student group members agree they need technical help during the experiment and two of them do not think they need to learn much before using the system. Considering that the research students can get help during the experiment and all of them think they need help during the experiment, we can find that sufficient help or training are necessary at the beginning of using the system.

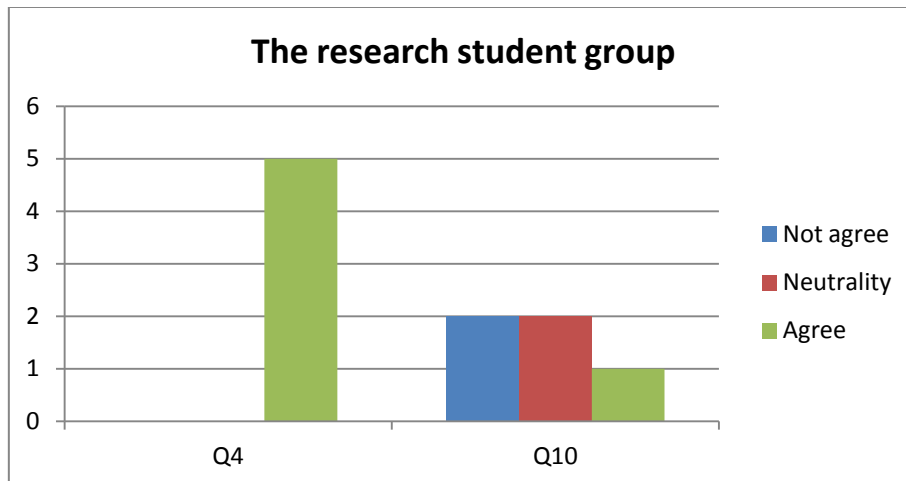


Figure 5-20 The opinions about learnability of the research student group

Based on the above discussion, we can find that the participants need helps when they use the system but they can handle it when they are familiar with it. This dashboard allows the participants to create indicators so that it is more complicated than most of the dashboards which supply standard and predefined indicators. In order to reduce the difficulties and help the novices, it could be useful to propose 2 or 3 classical predefined indicators in DDART so that the novices can visualize them directly and understand how the customizable indicators are created. When they have experiences, they can use the dashboard to create their own indicators. Furthermore, it is necessary to supply users with sufficient helps or tutorials before they use the dashboard, even though two tutorial videos have already been supplied. Especially, it will be much more helpful if a tutorial course can be given face to face before users use the system.

Readability

Before the experiment, we have created four indicators in the DDART system. In the scenario, eleven questions were supplied to the participants after they observed these predefined indicators. For example, they were asked who is the most active in the group, how long is your weekly average work time etc. We calculated the average accuracy of the answers (A) by the following formula:

$$A = \frac{\sum_{i=1}^N (C_i/Q)}{N}$$

Where: N is the group size, Q is the quantity of the questions ($Q=11$), C_i is the quantity of the correct answers given by participant i . C_i is calculated based on the answers in the survey.

Table 5-6 The average accuracy of the answers in survey

Engineering student group	Research student group
90.11%	90.77%

From the above table, the average accuracies of the two groups are almost the same, which are up to 90%. It indicates that the readability of the indicators created in the DDART is remarkable. Namely, most of the participants can extract the right information from the indicators and the indicators can be understood by most of the participants.

5.2.5.2 Utility

Which indicator is useful?

In the survey, we ask the participants to select the indicators, which seem useful according to their previous project experiences, from eight indicators:

1. Peer assessments (the judgments you get from the other group members)
2. Social network (the social interactions between the group members)
3. Daily work time (the time you spent on the project every day)
4. Scheduled vs real (comparison between the schedule of a task and its real progresses)
5. System login duration (the time spent in the system)
6. Tool used frequency (the frequencies of the tools used by some group members)
7. Activity frequency (the frequencies of some activities done by some group members)
8. Progress in learning (comparison knowledge/skill progress with its target level)

Figure 5-21 is the result of the selections. The Y axe refers to indicator id we list above and the X axe represents the frequency of each indicator selected by participants.

75% participants selected the indicator “*Scheduled vs real*”. We can find most of the participants think it is useful to compare their project schedules with the actual progresses. Hence, we highlight that it is important to pay more attention to the planification process in PBL and help learners to carry out the project according to the schedules in PBL. In the future research, it will be useful to provide more related indicators to learners, such as the calendar indicator which can remind learners that a specific task is required to be accomplished in

one week. Furthermore, the system could supply learners with more powerful planification functions.

The “*Peer assessments*” indicator and “*Social network*” indicator are following behind (58%). We can find that half of the participants pay attention on their social states in the group. They concern how the others evaluate them and how they interact with the others. Hence, in the future research, we have to focus to assessment aspect in PBL. In DDART, we offer five likert scales (very good/good/neutral/not very good/not at all good) to learners to assess their peers from a general point of view. We can subdivide these five scales into finer scales or supply the function which enables learners to assess their peers from different aspects (communication, creativity, execution etc.) rather than globally. So learners can observe how their peers evaluate them in different aspects. DDART calculate the indicator “*social network*” based on the interaction frequency. In the future research, it is better to consider more factors together, for example, the interaction time. Especially, the two social indicators can be combined together. When learners observe the interactions between themselves and the others, they also can view how the others evaluate them at the same time.

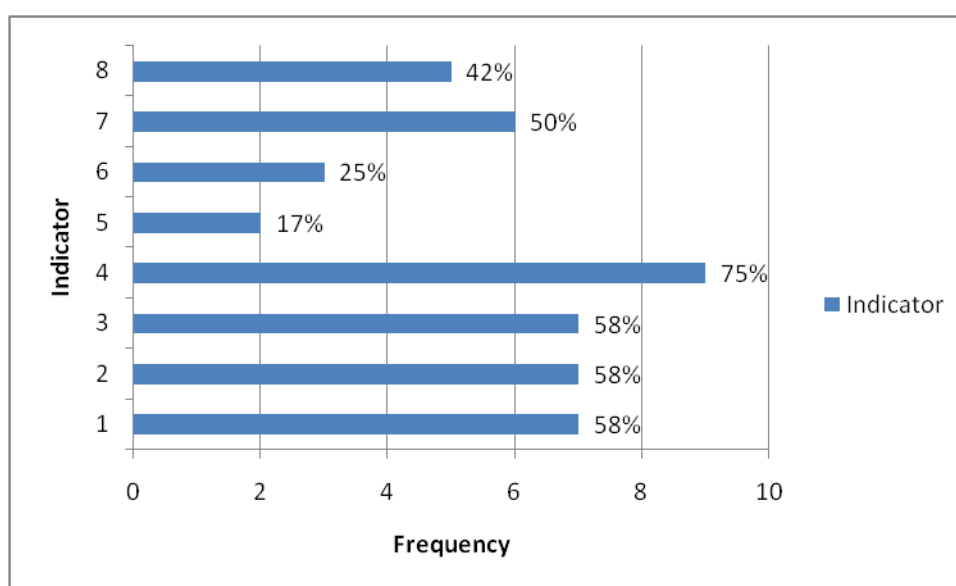


Figure 5-21 Which indicator is useful?

Is DDART useful in managing project?

We analyzed the feedbacks of the surveys and classify the comments.

1 Seven participants affirmed that DDART is useful in the project because it allows them to create indicators to monitor their work.

Participant 1: “*This is useful for collective project because it is a personalized dashboard. It offers infinite possibilities...it is easy to extract any information from the data, which is a very good point*”.

Participant 2 (translated from French): “*I think it is useful because we can ask the questions about the resources and tasks... It tracks the time spent by*

everyone on a specific task and then communicate it with his group. And it is explicit for everyone and provides helps in some cases."

Participant 3 (translated from French): *"How the others judge me? What is the social network? I find this information is interesting to know...It does not need to code (for example, the macro or complicated formulas in Excel)".*

Participant 4: *"I think it will be useful for users to evaluate his work and collaborate with other team members. It can help me to record the time nodes of tasks and follow my work and keep aware of other's progress."*

Participant 5 (translated from French): *"The object of this system is very good (analyze the time, the tasks, etc.). It is useful because we can create indicator freely"*.

Participant 6 (translated from French): *"It allows to create personal indicators which is not permitted by using a team dashboard."*

Participant 7 (translated from French): *"It is helpful because we can create indicators and modify them without need to learn another special language...It is simpler than Excel and the social network is integrated."*

2 The other five participants proposed some aspects which restrict the usage of DDART.

Participant 1 (translated from French): *"If the phrases are not prepared, it could be simpler to input the task information."*

Participant 2 (translated from French): *"I think that my previous dashboard is more appropriate to input information like abilities and tasks."*

Participant 3 (translated from French): *"The most difficult is to unite the group members to input the information and follow the indicators."*

Participant 4 (translated from French): *"There are many possibilities to create indicators. But most indicators require to input information manually. The more information to input, the more difficult to manage indicators. The system should be more attractive and easier to use."*

Participant 5 (translated from French): *"It depends on the projects. The dashboard is not suitable for a short project with few people."*

From the above comments, we can find that DDART is useful in the projects, which can enable learners to follow and monitor their projects by observing personalized indicators. The learners in the projects can observe these indicators so as to adjust their behaviors or learning strategies. For example, the indicator "social network" can present learners' social states in the group; the indicator "daily work time" can help learners to adjust the time allocation; the indicator "peer assessment" enables learners to know how the others evaluate them. Furthermore, DDART allows learners to create customizable indicators. Learners can organize and plan their own ways to monitor the projects, which offers great opportunities to learners to develop self-regulation skills.

However, some participants also proposed some limitations of DDART used in PBL. The main limitation is the way to input the information. They think

the semi-structured sentences restrict them to write freely and it is not convenient to input information manually.

5.3 Conclusion

In this chapter, we presented two kinds of test of the performance of DDART. At first, we created some indicators with DDART, which are proposed in existing researches. We showed that DDART can create customizable indicators regarding to the activity, social, affective and cognitive categories. It indicates that the features of DDART are valid.

In the second part, we tested the usability and the utility of DDART by launching an experiment. By analyzing the experiment traces and the feedbacks of the surveys, we concluded some advantages and disadvantages of DDART. The reporting tool supports reflection and enables learners to input information to create indicators. The semi-structured sentence models give learners the possibility to input individual structured information which can be exploited. However, it takes much time for learners to input information manually. The semi-structured sentence model restricts learners to record freely, which affects the flexibility of DDART.

The dashboard allows learners to create customizable indicators. Most of the users can understand the indicators and extract the important information from the visualizations of the indicators. However, it is a challenge for the novices to use DDART to create indicators at the beginning. It is necessary to supply them with more help before they use the dashboard. The user experience should be enhanced by improving the interface design and accelerating the application loading. The time of the experiment was short, so most participants thought DDART is not easy to master in such a short time. Hence, it will be better to do another long term experiment to collect more precise observations from the participants. Most of the participants think the “*Scheduled vs real*”, “*Peer assessments*” and “*Social network*” indicators are the most useful indicators during the project. It emphasizes that we have to pay more attention to the planification and assessment processes in PBL.

6 Conclusion and Perspectives

6.1 Conclusion

In this research, we mainly studied the main systems used to help learners to gain self-regulation skills in Project-based Learning. We proposed to help learners to reflect on their learning and projects by recording their activities, assessments and judgments manually (self-report). Furthermore, by integrating the activity traces and the reporting traces, we give learners the possibility to explore integrated traces and so to create customizable indicators. This process helps them to enhance their skills in planification, reflection, self-monitoring and assessment in PBL.

Firstly, we proposed a system framework, called PBLMS (Project-based Learning Management System), which is composed of four modules:

1. **Trace collection.** Regarding to the different features of the traces, two types of traces are distinguished: activity traces and reporting traces. Activity traces record automatically information on how the learners interact with the system. Reporting traces record learners' reflections which contain important information that cannot be captured by the system. We designed different methods and data models to collect and store them. Collecting sensors are proposed to capture the activity traces automatically when learners do the different tasks with the PBLMS. We also designed a reporting tool that learners can use to report the activities carried out of the PBLMS and their reflections according to predefined semi-structured sentences.
2. **Trace integration.** In order to integrate the two types of traces and to create indicators based on them, we proposed an integrated traces model. This model can describe the integrated traces from different aspects: the activity, the learner, the time, the place and the tool. The reporting traces and the activity traces are structured according to this model to generate the integrated traces. The integrated traces are the data sources of the indicator creation. Learners can explore the integrated traces from different aspects by selecting different entities to focus on.
3. **Indicator calculation.** At first, we specified two data selection methods: crossing and filtering. Crossing can calculate the Cartesian product of two entity value sets and filtering can remove some traces from the integrated trace. Learners can use these two methods to focus on the traces they want to explore. Then, we propose five data types which show the qualitative and quantitative values from the traces that have been crossed and filtered: Frequency, Time interval, Time span, Content, and Description. At last, we designed calculation operators which help learners to edit mathematic formulas to be applied on the entity values: addition, subtraction, multiplication and division.
4. **Indicator visualization.** Different visualizations are finally provided to allow learners to observe the indicators from different points of view. We proposed a

process to present the indicators with different diagrams: selection of the visualization, adjustment of the data format, and indicator presentation.

Secondly, based on the above framework and methods, we developed a web based PBLMS, which integrates the Moodle LMS and DDART, a dynamic dashboard based on integrated traces. We consider that the four modules Chat, Forum, Wiki, and Resource of Moodle can help learners to carry out their projects. The activities carried out in these four modules have been collected as activity traces by the system sensors. DDART is composed of a reporting tool and a dynamic dashboard. The reporting tool supports learners' self-reporting according to a semi-structured sentences model. It helps learners not only to record their activities but also to reflect on their behaviors. The dynamic dashboard integrates the activity and reporting traces and enables learners to explore the integrated traces in order to create customizable indicators by dragging and dropping the parameters. It also offers eleven visualization modes for learners to present the indicators. The calculation process is WYSIWYG (what you see is what you get) so that learners can get the results without delay.

At last, in order to test the features of DDART, we firstly created various kinds of indicators which are proposed in existing researches for the analysis of activities, cognition, emotion and social state. Even though DDART has some limitations, we can validate the system because it can create most of existing indicators. Then, in order to test the usability and perceived utility, we conducted an experiment with twelve participants. From the experiment, we can find that the reporting tool supports reflection and the functions are easy to use even if the participants have to spend much time on writing reports. The dashboard supports learners to monitor and self-regulate their behaviors because learners can create personalized indicators. But it could be hard for the novices to create indicators in the dashboard. The indicators presented in the dashboard are readable and learners can extract important information from them. In the further research, it is necessary to improve DDART based on the experiment results, for example by accelerating the application loading and by replacing some ambiguous words.

6.2 Perspectives

Regarding the future works, we put forward some proposals from several aspects:

1. *To carry out a long term experiment.*

At this step of the research work, we conducted a short and semi-controlled experiment because the time was limited and the quality of DDART was not sufficient enough to be used in real learning context. So we tested the perceived utility of DDART. We need to carry out a long-term experiment to test its utility in a real PBL course. Complementary studies have to be done in order to find, or adapt, some PBL situations that include Moodle and DDART.

2. *To support tutors to gain self-regulation skills*

In our research, DDART is developed from the learners' points of views and we focus on how to support learner's self-regulation learning in PBL. However, tutors' self-regulation is ignored by most existing researches. It is also important in PBL

because tutors have to monitor the process of PBL and to help learners to reach their learning goals independently and consciously. Considering the differences between learner's self-regulation and tutors' self-regulation, it is necessary to develop a system for tutors to improve their self-regulation skills. Several research questions need to be considered, such as "*How to help tutors to reflect on their efforts to monitor PBL?*", "*Which types of indicators are useful for tutors to monitor their own behaviors in order to help learners to obtain SRL skills?*", "*How to lead tutors to share their PBL experiences?*"

3. To improve the methods of inputting reporting traces.

From the experiment results, we observe that most of the participants think the reporting tool takes much time for writing reports because they have to input the information according to the semi-structured sentence models instead of writing freely. So we can enable learners to write freely by importing the Natural Language Processing (NLP) technics into the reporting tool. NLP enables computers to extract information from natural language input. It is a complex research issue because it relates with the fields of linguistics, computer science, artificial intelligence, machine learning and so on. Many challenges in NLP are identified, such as spelling correction, information extraction and parsing, and language translation, topic recognition. After NLP is applied in the reporting tool, the system can extract the important information from the learners' reports and then generate the reporting traces to store into the database.

Another solution is to take advantage of a vocal assistant like Siri (developed by Apple Inc.). This is a simple and direct way for learners to report information without constraints. They can report anywhere and at anytime with their smartphones. The system can translate their vocal information into storable texts and generate reporting traces. Plenty of research questions have to be considered, such as "*How is DDART compatible with different vocal assistants in different operation systems (Android, iOS, windows)?*"; "*How to upload the vocal information into the Cloud?*"; "*How to process the vocal data and extract information?*"; "*Which data model can transform the vocal information into reporting traces?*"; "*How to store the reporting traces?*"

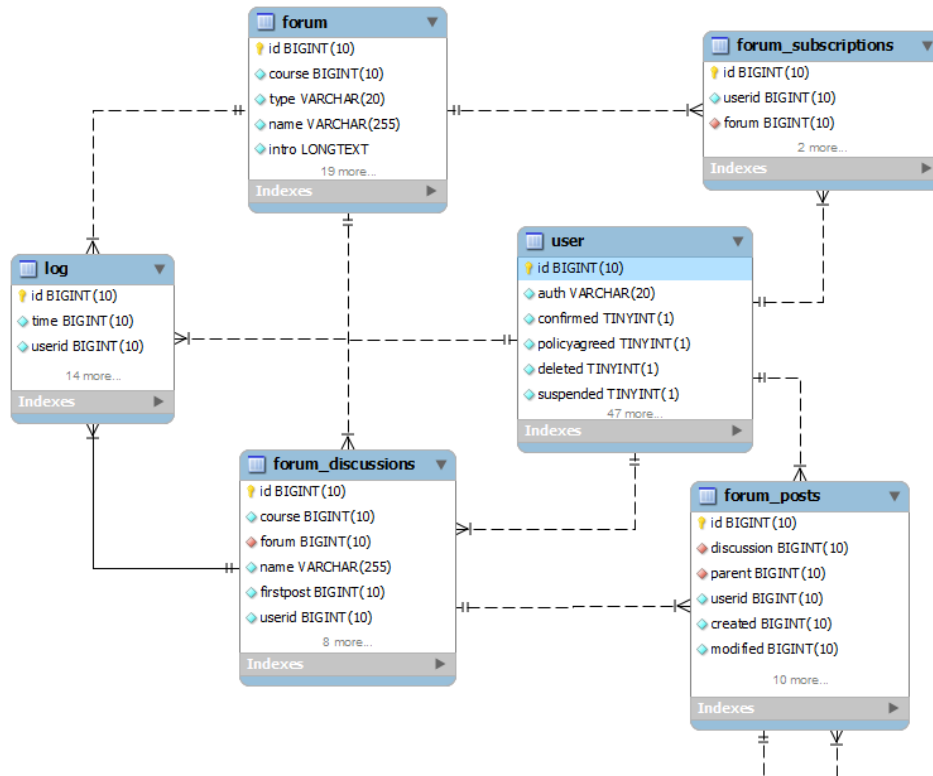
4. To analyze the indicator results and to give advices automatically.

In order to help learners to extract the information from the indicators, it will be helpful if DDART can interpret the indicators and give to learners the appropriate suggestions based on the indicator results. For example, if the indicator shows a learner spends much time in chatting and little time in his programming tasks, DDART can give an alert to this learner: "*You spent too much time on chatting. Pay more attention of programming.*" DDART can analyze several indicators together, for example an indicator that shows that a learner is depressed because he has a struggle to programming in JAVA and another indicator that shows that Julien spent much time on programming in JAVA. DDART can suggest him: "*Contact and do the activities with Julien, he has skills in JAVA programming.*" In order to realize this function, it could be helpful to integrate an expert system into DDART which can give expert opinions to learners. The knowledge base is the kernel of the expert system, which includes plenty of facts and rules. It is necessary to consider how to extract facts and generate rules automatically based on the indicators created by learners.

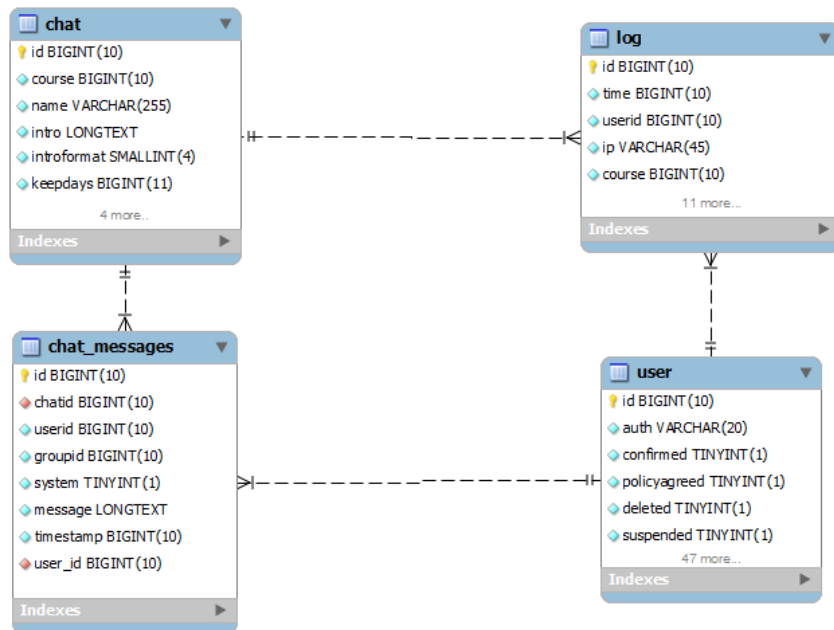
Annex

A. The data models of the activity traces in Moodle tools

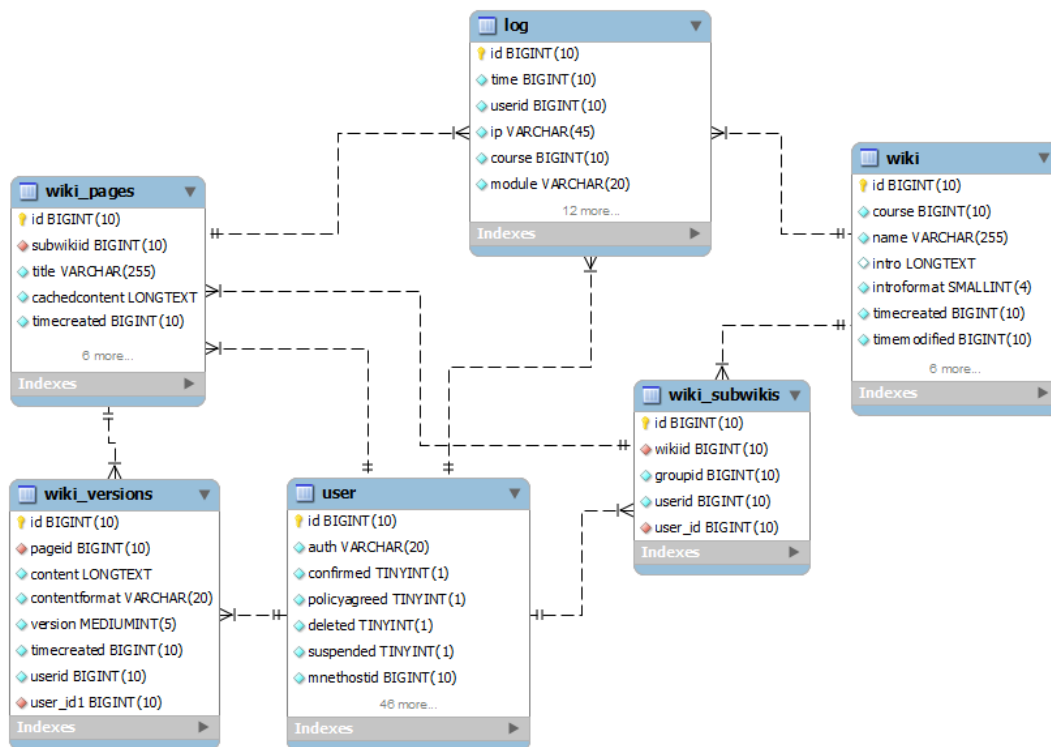
1. The data model of Forum



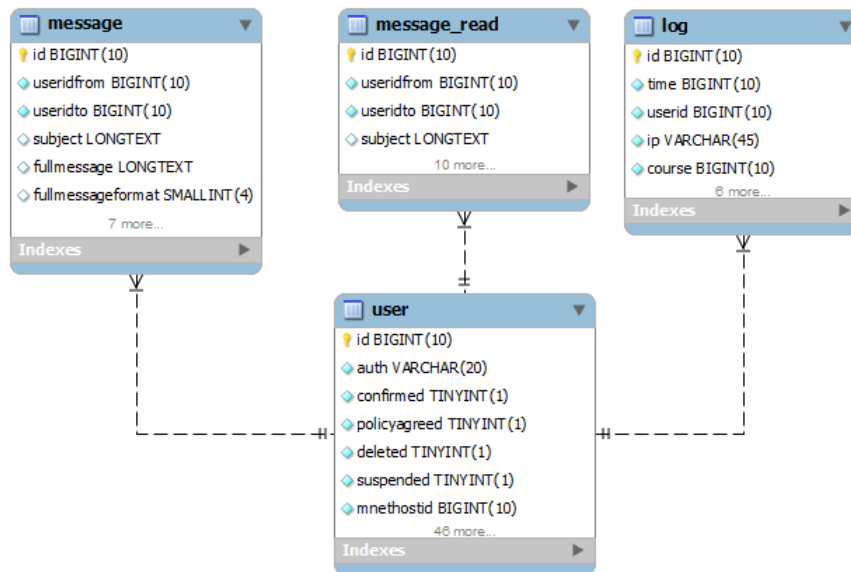
2. The data model of Chat



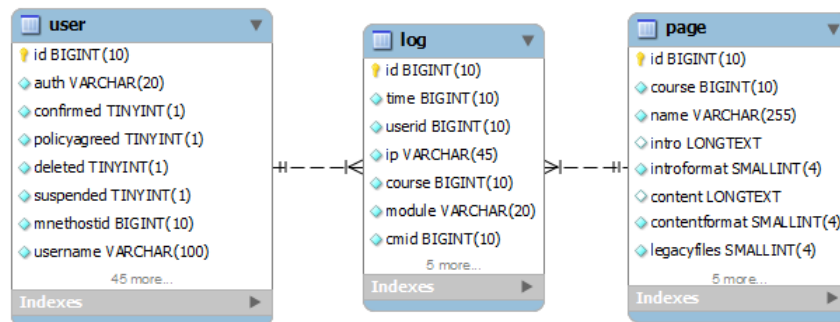
3. The data model of Wiki



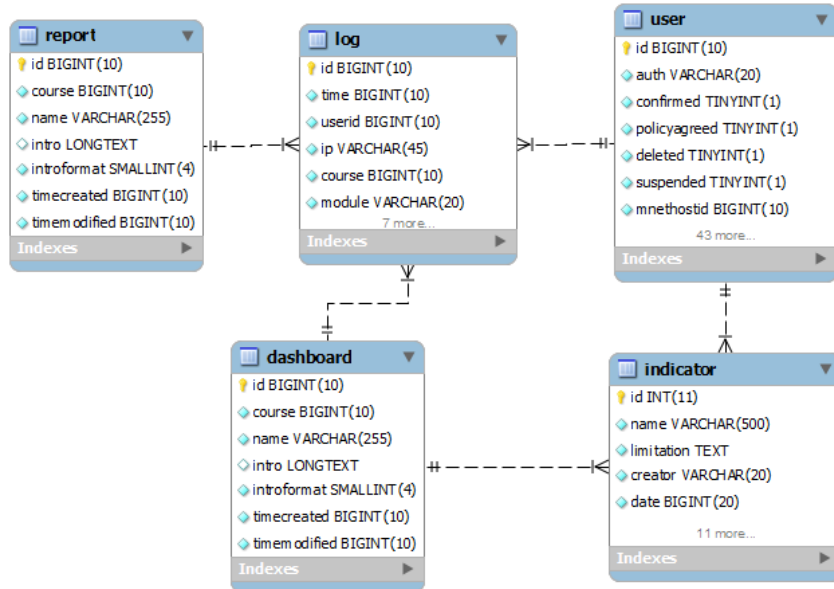
4. The data model of system connection



5. The data model of Resources



B. The data models of the activity traces in reporting tool and dashboard



C. The DTD and XML used in reporting tool

1. The DTD of Category class and the XML file

```
<?xml version="1.0" encoding="utf-8"?>
  <!DOCTYPE categoryys[
    <!ELEMENT categoryys (category*)>
    <!ELEMENT category (name)>
  <!ELEMENT name (#PCDATA)>
  <!ATTLIST category
    id ID #REQUIRED
    creatorId CDATA #REQUIRED
    createTime CDATA #REQUIRED>
  ]>
```

```
<categoryys>
  <category id="1" creatorId="2" createTime="07/11/2012">
    <name>project tasks</name>
  </category>
  <category id="2" creatorId="2" createTime="07/11/2012">
    <name>self-assess</name>
  </category>
  <category id="3" creatorId="2" createTime="07/11/2012">
    <name>judgement</name>
  </category>
  <category id="4" creatorId="2" createTime="07/11/2012">
    <name>plan</name>
  </category>
</categoryys>
```

2. The DTD of Variable Type class and the XML file

```
<?xml version="1.0" encoding="utf-8"?>
  <!DOCTYPE variableTypes[
    <!ELEMENT variableTypes (variableType*)>
    <!ELEMENT variableType (name)>
  <!ELEMENT name (#PCDATA)>
  <!ATTLIST variableType
    id ID #REQUIRED>
  ]>
```

```
<variableTypes>
  <variableType id="1">
    <name>drop down list</name>
  </variableType>
  <variableType id="2">
    <name>label</name>
  </variableType>
  <variableType id="3">
    <name>calendar</name>
  </variableType>
</variableTypes>
```


3. The DTD of Report Structure class and the XML file

```
<?xml version="1.0" encoding="utf-8"?>
  <!DOCTYPE reportStructures[
    <!ELEMENT reportStructures (reportStructure*)>
    <!ELEMENT reportStructure (sentenceid+)>
  <!ELEMENT sentenceid (#PCDATA)>
  <!ATTLIST reportStructure
    id ID #REQUIRED
    addtime CDATA #REQUIRED>
  ]>
```

```
<reportStructures>
  <reportStructure id="1" addtime="01-04-2013">
    <sentenceid>2</sentenceid>
    <sentenceid>3</sentenceid>
  </reportStructure>
  <reportStructure id="2" addtime="01-04-2013">
    <sentenceid>1</sentenceid>
    <sentenceid>5</sentenceid>
    <sentenceid>6</sentenceid>
  </reportStructure>
</reportStructures>
```

4. The DTD of Variable Value class and the XML file

```
<?xml version="1.0" encoding="utf-8"?>
  <!DOCTYPE variableValues[
    <!ELEMENT variableValues (variableValue*)>
    <!ELEMENT variableValue (value)>
  <!ELEMENT value (#PCDATA)>
  <!ATTLIST variableValue
    id ID #REQUIRED>
  ]>
```

```
<variableValues>
  <variableValue id="1">
    <value>very good</value>
  </variableValue>
  <variableValue id="2">
    <value>good</value>
  </variableValue>
  <variableValue id="3">
    <value>neutral</value>
  </variableValue>
  <variableValue id="4">
    <value>not very good</value>
  </variableValue>
  <variableValue id="5">
    <value>not at all good</value>
  </variableValue>
</variableValues>
```

5. The DTD of Customized Structure class and the XML file

```
<?xml version="1.0" encoding="utf-8"?>
  <!DOCTYPE acts[
    <!ELEMENT acts (user*)>
    <!ELEMENT user (reportStructureid+)>
    <!ELEMENT reportStructureid (#PCDATA)>
    <!ATTLIST reportStructureid
      courseid CDATA #REQUIRED
      type CDATA #REQUIRED>
    addtime CDATA #REQUIRED>
  ]>
```

```
<acts>
  <user id="2">
    <reportStructureid courseid="2" type="plan"
      addtime="01-04-2013">1</reportStructureid>
    <reportStructureid courseid="2" type="activity"
      addtime="05-04-2013">2</reportStructureid>
    <reportStructureid courseid="3" type="plan"
      addtime="01-04-2013">1</reportStructureid>
    <reportStructureid courseid="3" type="activity"
      addtime="01-04-2013">2</reportStructureid>
  </user>
</acts>
```

6. The DTD of Report Content class and the XML file

```
<?xml version="1.0" encoding="utf-8"?>
  <!DOCTYPE reports[
    <!ELEMENT reports (report*)>
    <!ELEMENT report (content+, comments*)>
    <!ELEMENT content (sentence+)>
    <!ELEMENT sentence (variable+)>
    <!ELEMENT variable (value+)>
    <!ELEMENT value (#PCDATA)>
    <!ELEMENT comments (comment*)>
    <!ELEMENT comment (#PCDATA)>
    <!ATTLIST report
      id ID #REQUIRED
      writerId CDATA #REQUIRED
      title CDATA #REQUIRED
      createTime CDATA #REQUIRED
      updateTime CDATA #IMPLIED
      privacy (public|private) #REQUIRED
      courseid CDATA #REQUIRED>
    <!ATTLIST variable id ID #REQUIRED>
    <!ATTLIST sentence id CDATA #REQUIRED>
    <!ATTLIST comment
      Id ID #REQUIRED
      previousId IDREF #REQUIRED
      writerId CDATA #REQUIRED
      createTime CDATA #REQUIRED>
    content CDATA #REQUIRED
  ]>
```

```
<reports>
  <report id="1" courseid="2" writerId="3" title="goal report"
    createTime="12-1-2014 00:17:40" updateTime="no"
    privacy="public">
    <content>
      <sentence id="3">
        <variable id="1">
          <value>organize a meeting</value>
        </variable>
        <variable id="2">
          <value>11-01-2014 08:00</value>
        </variable>
        <variable id="3">
          <value>11-01-2014 08:20</value>
        </variable>
      </sentence>
      <sentence id="2">
        <variable id="1">
          <value>mysql</value>
        </variable>
        <variable id="2">
          <value>good</value>
        </variable>
      </sentence>
      <sentence id="2">
        <variable id="1">
          <value>documentation</value>
        </variable>
        <variable id="2">
          <value>good</value>
        </variable>
      </sentence>
    </content>
    <comments>
      <comment1 id="1" previousId="" writerId="5"
        createTime="12-1-2014 18:21:04" content="good report!" />
    </comments>
  </report>
</reports>
```

7. The DTD of Semi-structured sentence class and the XML file

<pre> <?xml version="1.0" encoding="utf-8"?> <!DOCTYPE sentences[<!ELEMENT sentences (sentence*)> <!ELEMENT sentence (part+, variable+)> <!ELEMENT variable (valueId*)> <!ELEMENT valueId (#PCDATA)> <!ELEMENT part (#PCDATA)> <!ATTLIST sentence id ID #REQUIRED categoryId CDATA #REQUIRED createTime CDATA #REQUIRED creatorId CDATA #REQUIRED> <!ATTLIST variable id ID #REQUIRED typeId CDATA #REQUIRED value CDATA #REQUIRED > <!ATTLIST part id ID #REQUIRED>]> </pre>	<pre> <sentences> <sentence id="1" categoryId="4" creatorId="2" createTime="07/11/2012"> <part id="1">The</part> <variable id="1" typeId="1" value="no"> <valueId>6</valueId> <valueId>7</valueId> </variable> <part id="2">is to</part> <variable id="2" typeId="2" value="do what"/> <part id="3">from</part> <variable id="3" typeId="3" value="16-01-2012 00:00"/> <part id="4">to</part> <variable id="4" typeId="3" value="16-01-2012 23:59"/> </sentence> <sentence id="2" categoryId="4" creatorId="2" createTime="07/11/2012"> <part id="1">My</part> <variable id="1" typeId="2" value="object(skill, knowledge.. <part id="2">should reach the level of</part> <variable id="2" typeId="1" value="no"> <valueId>1</valueId> <valueId>2</valueId> <valueId>3</valueId> <valueId>4</valueId> <valueId>5</valueId> </variable> </sentence> <sentence id="3" categoryId="4" creatorId="2" createTime="07/11/2012"> <part id="1">I schedule to</part> <variable id="1" typeId="2" value="do what"/> <part id="2">from</part> <variable id="2" typeId="3" value="16-01-2012 00:00"/> <part id="3">to</part> <variable id="3" typeId="3" value="16-01-2012 23:59"/> </sentence> </pre>
---	---

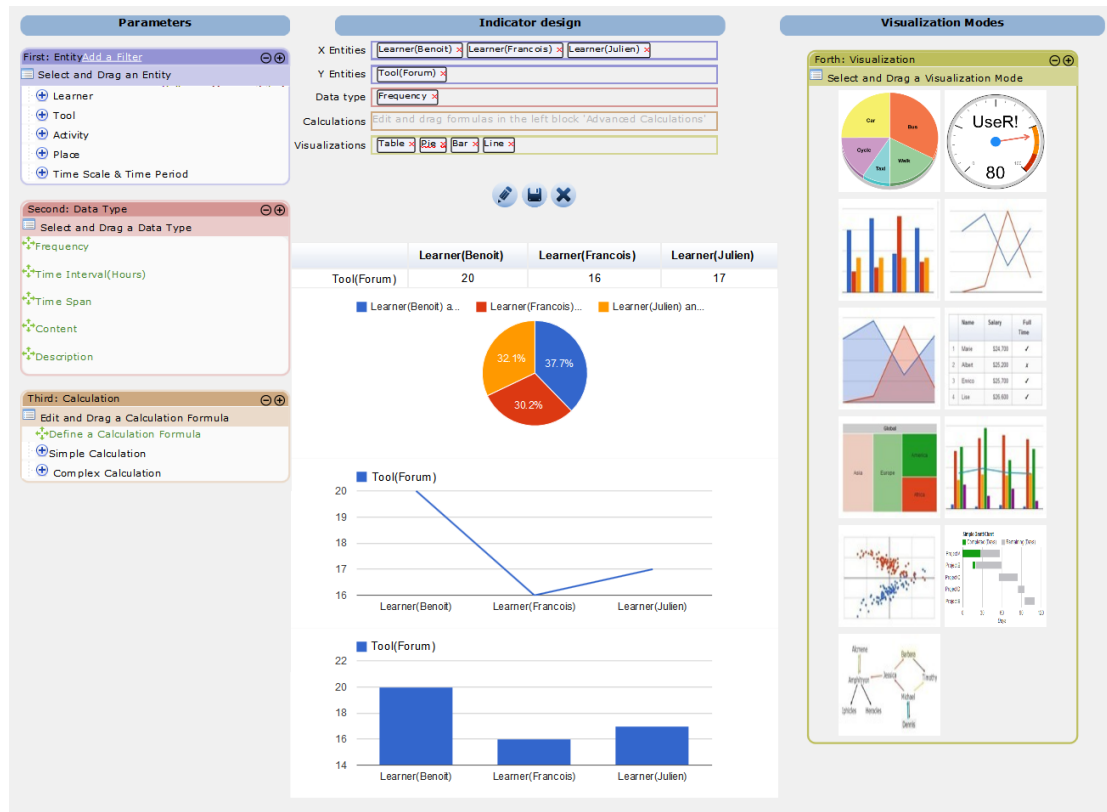
```

<sentence id="4" categoryId="1" creatorId="2" createTime="08/04/2013">
  <part id="1">I</part>
  <variable id="1" typeId="2" value="do what(i.e talk)"/>
  <part id="2">with</part>
  <variable id="2" typeId="2" value="whom(i.e tom, mary)"/>
  <part id="3">about</part>
  <variable id="3" typeId="2" value="what (i.e java)"/>
  <part id="4">using</part>
  <variable id="4" typeId="2" value="tool name(i.e skype)"/>
  <part id="5">from</part>
  <variable id="5" typeId="3" value="16-01-2012 00:00"/>
  <part id="6">to</part>
  <variable id="6" typeId="3" value="16-01-2012 23:59"/>
</sentence>
<sentence id="5" categoryId="3" creatorId="2" createTime="07/11/2012">
  <part id="1">I judge that</part>
  <variable id="1" typeId="2" value="object (tom, a book..)"/>
  <part id="2">is</part>
  <variable id="2" typeId="1">
    <valueId>1</valueId>
    <valueId>2</valueId>
    <valueId>3</valueId>
    <valueId>4</valueId>
    <valueId>5</valueId>
  </variable>
  <part id="3">because</part>
  <variable id="3" typeId="2" value="reason (he is nice)"/>
</sentence>
<sentence id="6" categoryId="2" creatorId="2" createTime="07/11/2012">
  <part id="1">I self assess that my</part>
  <variable id="1" typeId="2" value="object( skill, knowledge, mood..)"/>
  <part id="2">is</part>
  <variable id="2" typeId="1">
    <valueId>1</valueId>
    <valueId>2</valueId>
    <valueId>3</valueId>
    <valueId>4</valueId>
    <valueId>5</valueId>
  </variable>
  <part id="3">because</part>
  <variable id="3" typeId="2" value="reason (I am happy)"/>
</sentence>
</sentences>

```

D. Examples of visualization modes

1. Examples of “Table”, “Pie chart”, “Line chart”, “Bar chart”
(from top to bottom)



2. Examples of “Area chart”, “Tree map”, “Combo chart” (from top to bottom)

The screenshot displays a software interface for configuring data visualizations, divided into three main sections: Parameters, Indicator design, and Visualization Modes.

Parameters:

- First: Entity Add a Filter:** Includes a list of entities: Learner, Tool, Activity, Place, and Time Scale & Time Period.
- Second: Data Type:** Includes a list of data types: Frequency, Time Interval (Hours), Time Span, Content, and Description.
- Third: Calculation:** Includes options for Simple Calculation and Complex Calculation.

Indicator design:

- X Entities:** Tool(Forum), Tool(Wiki)
- Y Entities:** Learner(Benoit), Learner(Francois), Learner(Julien), Learner(Pascal)
- Data type:** Frequency
- Calculations:** Edit and drag formulas in the left block "Advanced Calculations"
- Visualizations:** Area, Combo, Treemap

Visualization Modes:

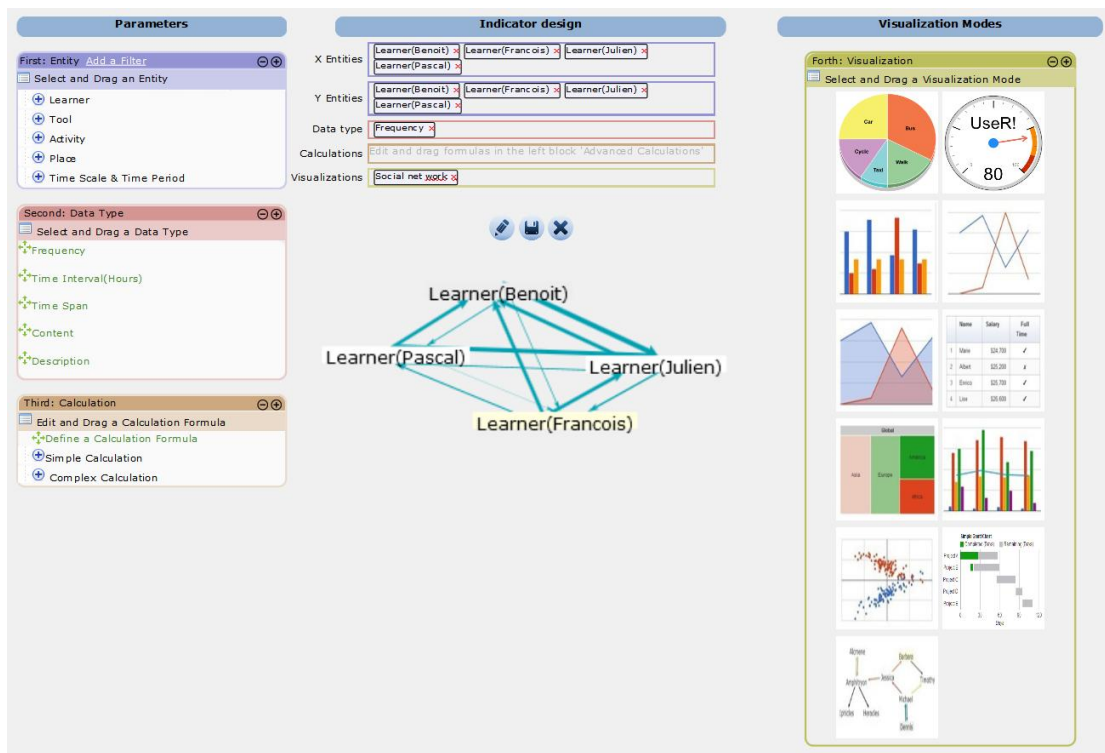
- Forth: Visualization:** Select and Drag a Visualization Mode. Includes a pie chart, a gauge, a bar chart, a line chart, a table, a treemap, a scatter plot, and a network diagram.

Area Chart (top): Shows the frequency of tool usage for four learners across two tools. The Y-axis ranges from 8 to 20. The X-axis shows Tool(Forum) and Tool(Wiki). The legend includes Learner(Benoit) (blue), Learner(Francois) (red), Learner(Julien) (orange), and Learner(Pascal) (green).

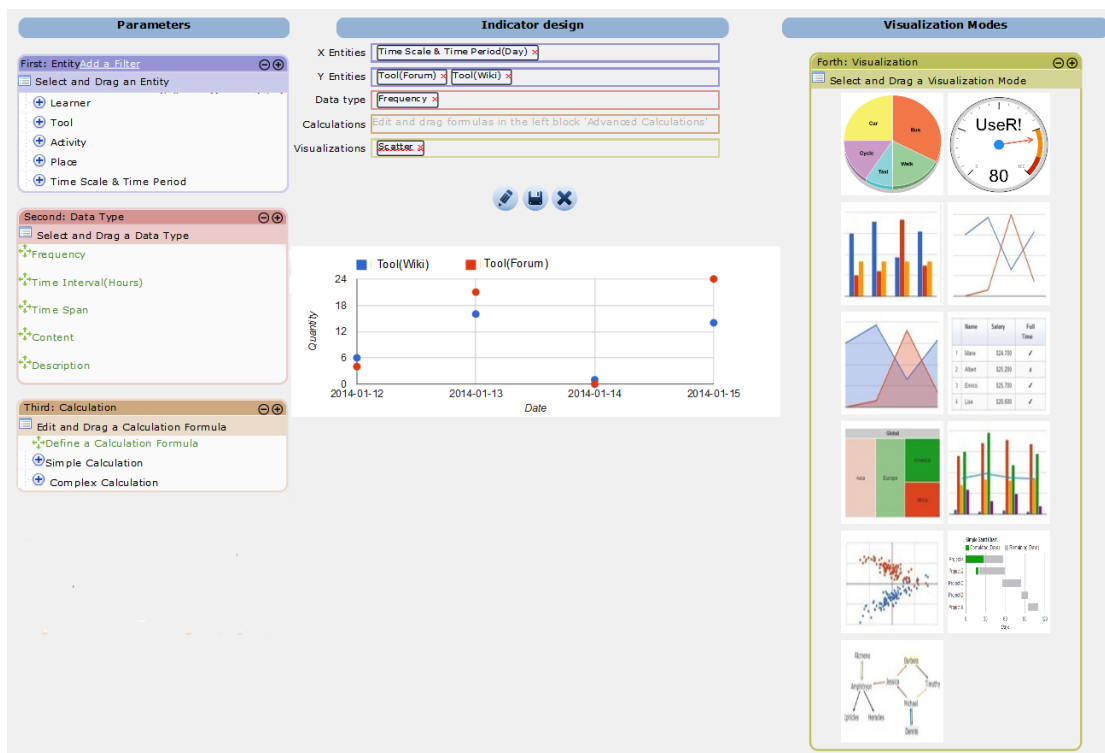
Treemap (middle): Titled "tool(Forum)", it shows the distribution of tool usage for four learners. The legend includes Learner(Benoit) (green), Learner(Julien) (light green), Learner(Francois) (grey), and Learner(Pas.) (red).

Combo Chart (bottom): Shows the frequency of tool usage for four learners across two tools. The Y-axis ranges from 8 to 20. The X-axis shows Tool(Forum) and Tool(Wiki). The legend includes Learner(Benoit) (blue), Learner(Francois) (red), Learner(Julien) (orange), and Learner(Pascal) (green).

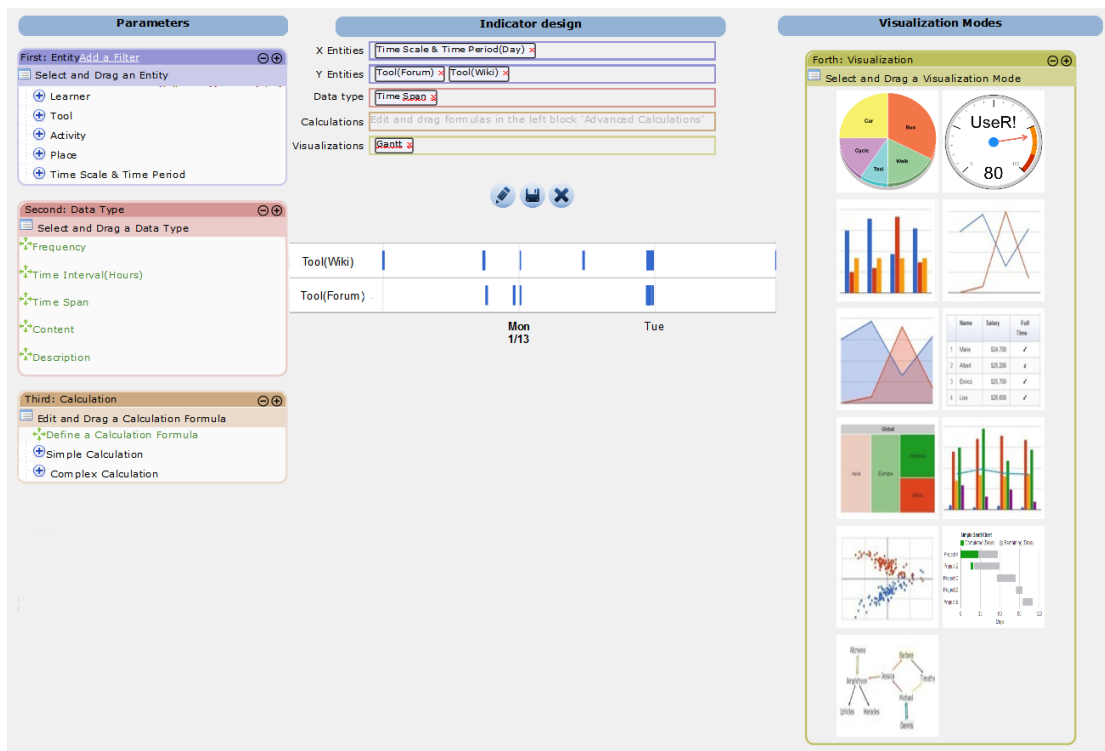
3. An example of “Social network chart”



4. An example of “Scatter chart”



5. An example of “Gantt chart”



6. An example of “Gauge chart”

The screenshot displays a software interface for configuring a gauge chart. It is organized into three main panels:

- Parameters:**
 - First: Entity Add a Filter:** Includes options for Learner, Tool, Activity, Place, and Time Scale & Time Period.
 - Second: Data Type:** Includes options for Frequency, Time Interval (Hours), Time Span, Content, and Description.
 - Third: Calculation:** Includes options for Simple Calculation and Complex Calculation.
- Indicator design:**
 - X Entities:** Learner(Benoit), Learner(Francois)
 - Y Entities:** Tool(Forum), Tool(Chat)
 - Data type:** Frequency
 - Calculations:** Edit and drag formulas in the left block "Advanced Calculations"
 - Visualizations:** Gauge
- Visualization Modes:** A grid of various chart types including pie charts, bar charts, line graphs, a table, and a network diagram.

Below the Indicator design section, four gauge charts are shown with their respective labels:

- Gauge 1: 20, labeled "Tool(Forum) & Learner(Benoit)"
- Gauge 2: 16, labeled "Tool(Forum) & Learner(Francois)"
- Gauge 3: 19, labeled "Tool(Chat) & Learner(Benoit)"
- Gauge 4: 18, labeled "Tool(Chat) & Learner(Francois)"

E. The experiment scenario

Nous vous invitons à suivre le scénario ci-après afin de tester l'outil DDART qui comporte un tableau de bord et un outil de reporting. Selon les résultats de l'expérimentation, DDART pourrait être proposé l'année prochaine aux groupes de PCo et votre retour nous est précieux pour l'améliorer. Nous vous remercions par avance du temps consacré à ce test.

* Required

Votre nom * Votre prénom *

I. Contexte et rôle

Vous êtes Pascal, un membre d'un groupe impliqué dans un projet. L'objectif du projet est de développer un petit site Web commercial. Le projet se déroule du 11 au 30 janvier 2014. Aujourd'hui, nous sommes le 16 janvier 2014. Vous et les autres membres du groupe (Sophie, Julien, Benoit, François) avez fait certaines tâches du projet en 6 jours (par exemple : communiquer, partager des informations, faire une conférence, faire un brainstorming).

Vous avez en effet la possibilité d'utiliser la plate-forme d'apprentissage « Moodle » ou d'autres outils informatiques (google doc, dropbox, ...) pour réaliser votre projet. Pour savoir comment l'équipe a travaillé, vous pouvez suivre:

- 1) Les activités effectuées avec les outils informatiques dans la plate-forme Moodle (wiki, chat, forum, messages privés),
- 2) Les activités effectuées avec d'autres outils informatiques hors de la plate-forme Moodle (Skype, Firefox, Gmail, Word, ...)
- 3) Les activités effectuées sans outils informatiques (lire des livres, faire une réunion,...).

DDART peut enregistrer automatiquement les activités faites avec Moodle mais pas les autres. Pour les suivre, il faut donc les décrire en utilisant l'outil de reporting. Il est donc vraiment important de rapporter toutes les tâches que vous avez effectuées en dehors de Moodle.

L'outil de tableau de bord vous aide à créer les indicateurs de suivi du projet. La création d'indicateur est assez libre. Dans le cadre de l'expérimentation, il vous est demandé de créer des indicateurs pour suivre votre propre activité au cours du projet, les activités des autres membres du groupe et votre propre processus d'apprentissage.

II. Les vidéos de tutoriel

1. Regardez les vidéos de tutoriel d'explication du fonctionnement de l'outil de reporting et du tableau de bord.

Reporting tool: <https://www.youtube.com/watch?v=PC9OwFoDbAA>

Dashboard: https://www.youtube.com/watch?v=_yIYlyc9DIw

III. Tâches à réaliser

1. Cliquez sur le lien: ddart.fr/moodle (utilisez FireFox s'il vous plaît)
Login: Pascal
Mot de passe: Pascal_1
Cliquez sur le lien « Course 1 »

2. Entrez dans le module « ReportingTool » (ou « project report »). Plusieurs rapports sont stockés et correspondent aux activités antérieures du projet (voir l'onglet « Manage my reports »). Vous devez écrire un nouveau rapport et de l'enregistrer. Vous devez renseigner une tâche, un jugement sur vous et un jugement sur autre chose que vous. Nous vous laissons imaginer des tâches réalisées (comme par exemple écrire un e-mail, travailler avec un autre membre du groupe, programmer en PHP ou C++), des évaluations des autres membres du groupe (Sophie, Julien, Benoit, François) ou de vos compétences/connaissances (par exemple en programmation java, en compétence de planification).
3. Lisez un des rapports des autres membres du groupe et écrivez quelques commentaires sur ce rapport.
4. Entrez dans le module « Dynamic Dashboard ». Plusieurs indicateurs ont déjà été créés. Observez ces indicateurs et répondez aux questions suivantes:
 - 1) Qui est au centre du réseau social? *
 Julien Pascal Benoit Sophie Francois
 Qui est à la périphérie du réseau social? *
 Julien Pascal Benoit Sophie Francois
 Quelle est votre position dans le réseau social? *
 Centre entre le centre et la périphérie à la périphérie
 - 2) Quelle est la valeur moyenne des jugements que vous avez obtenus des autres? *

 Pourquoi les autres vous donnent-ils ces jugements? *

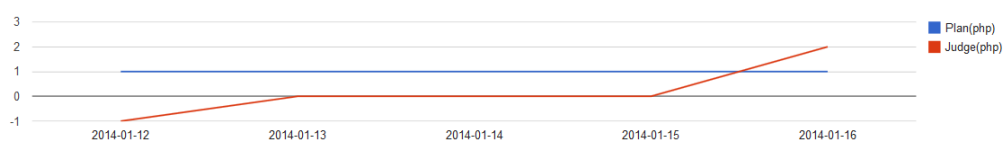
 - 3) Quel jour avez-vous passé le plus de temps à travailler pour le PCo? *

 Quel est votre temps moyen de travail? *

 - 4) Est-ce que votre tâche « read an ebook » a été effectuée selon votre calendrier prévu? *

 5. Au début du projet, vous avez défini un objectif en programmation PHP que vous souhaitez atteindre pour que le projet soit abouti. Avec l'avancement du projet, vous avez écrit des rapports pour enregistrer vos progrès en PHP avec 5 niveaux: pas du tout bon (-2), pas très bon (-1), neutre (0), bon (1), très bon (2). Maintenant, vous voulez comparer vos (Pascal) progrès avec votre niveau idéal pour savoir si vous avez atteint votre objectif ou non. La période de temps est du 12 au 17 Janvier 2014. Vous devez créer un indicateur sous la même forme que celui ci-dessous. Lorsque vous avez terminé, enregistrez cet indicateur en le nommant.

	2014-01-12	2014-01-13	2014-01-14	2014-01-15	2014-01-16
Plan(php)	1	1	1	1	1
Judge(php)	-1	0	0	0	2



Est-ce que vos compétences de programmation PHP atteint votre niveau prévu ? *

Oui Non

6. Vous voulez connaître la fréquence de plusieurs tâches du projet (écrire dans le wiki, écrire dans le chat, lire un ebook) effectués par vous-même (Pascal), par Julien et par Benoit. La période de temps est du 12 au 17 Janvier 2014. Créer cet indicateur. Lorsque vous avez terminé, enregistrez cet indicateur en le nommant. Quelles sont les tâches du projet que vous avez le plus effectués ? *

Quelles sont les tâches du projet que vous avez le moins effectués? *

F. The experiment survey

Nous vous remercions de bien vouloir répondre au questionnaire suivant en indiquant pour chaque phrase votre degré d'accord ou de désaccord (de 1: fortement en désaccord ; à 5 : entièrement d'accord).

* Required

1. J'aimerais utiliser ce système fréquemment *

fortement en désaccord

1	2	3	4	5
---	---	---	---	---

 entièrement d'accord

2. Je trouve ce système inutilement complexe *

fortement en désaccord

1	2	3	4	5
---	---	---	---	---

 entièrement d'accord

3. Je pense que ce système est facile à utiliser *

fortement en désaccord

1	2	3	4	5
---	---	---	---	---

 entièrement d'accord

4. J'aurais besoin d'un support technique pour pouvoir utiliser ce système *

fortement en désaccord

1	2	3	4	5
---	---	---	---	---

 entièrement d'accord

5. Les différentes fonctionnalités de ce système sont bien intégrées *

fortement en désaccord

1	2	3	4	5
---	---	---	---	---

 entièrement d'accord

6. Ce système est truffé d'incohérences *

fortement en désaccord

1	2	3	4	5
---	---	---	---	---

 entièrement d'accord

7. Les étudiants peuvent apprendre à utiliser ce système très rapidement *

fortement en désaccord

1	2	3	4	5
---	---	---	---	---

 entièrement d'accord

8. Ce système est lourd à utiliser *

fortement en désaccord

1	2	3	4	5
---	---	---	---	---

 entièrement d'accord

9. J'ai confiance en ce système *

fortement en désaccord

1	2	3	4	5
---	---	---	---	---

 entièrement d'accord

10. J'ai dû apprendre beaucoup de choses avant de pouvoir utiliser le système *

fortement en désaccord

1	2	3	4	5
---	---	---	---	---

 entièrement d'accord

11. Je pense que ce système est utile pour le module PCo(Projet Collectif) *

1	2	3	4	5
---	---	---	---	---

Merci d'expliquer votre avis *

12. Je pense que ce système pourrait m'aider à gérer et à suivre un projet (planification, gestion du temps et des tâches, ...) *

1	2	3	4	5
---	---	---	---	---

Merci d'expliquer votre avis *

13. Je pense que ce système pourrait m'aider à gérer et à suivre mes apprentissages et mes compétences *

1	2	3	4	5
---	---	---	---	---

Merci d'expliquer votre avis *

14. Dans l'avenir, si ce système m'est proposé, je voudrais l'utiliser *

1	2	3	4	5
---	---	---	---	---

Merci d'expliquer votre avis *

15. En comparaison avec les tableaux de bord que j'ai utilisé, je pense que ce système est plus utile et plus souple *

1	2	3	4	5
---	---	---	---	---

Merci d'expliquer votre avis *

16. Quels sont les points positifs du système? Merci d'expliquer votre avis *

Quels sont les points négatifs du système? Merci d'expliquer votre avis *

17. Veuillez indiquer les indicateurs qui vous semblent utiles *

- 1) Les jugements fait par les autres membres du groupe
- 2) Les interactions sociales entre les membres du groupe
- 3) Le temps de travail de chacun
- 4) Le calendrier prévu vs le niveau de réalisation des tâches
- 5) La durée de connexion au système
- 6) La fréquence d'utilisation des outils par les membres du groupe
- 7) La fréquence de réalisation des activités par les membres du groupe
- 8) Le niveau d'acquisition des connaissances et/ou de compétences par rapport au niveau cible

18. Avez-vous regardé les vidéos de tutoriel du reporting tool ? *

19. Avez-vous regardé les vidéos de tutoriel du tableau de bord ? *

Bibliography

- Abrami, P., and H. Barrett. 2005. "Directions for Research and Development on Electronic Portfolios." *Canadian Journal of Learning and Technology/La revue canadienne de l'apprentissage et de la technologie* 31(3).
- Albert, W., and T. Tullis. 2013. *Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics*. Newnes.
- Bangor, A., PT Kortum, and JT Miller. 2008. "An Empirical Evaluation of the System Usability Scale." *International Journal of Human-Computer Interaction* 24(6):574–94.
- Bell, S. 2010. "Project-Based Learning for the 21st Century: Skills for the Future." *The Clearing House* 83(2):39–43.
- Blumenfeld, PC, and E. Soloway. 1991. "Motivating Project-Based Learning: Sustaining the Doing, Supporting the Learning." *Educational psychologist* 26(3-4):369–98.
- Boaler, J. 1999. "Mathematics for the Moment, or the Millennium." *Education Week* 17(29):30–34.
- Bouhnik, D., and T. Marcus. 2006. "Interaction in Distance-learning Courses." *Journal of the American Society for Information Science and Technology* 57(3):299–305.
- Bratitsis, T., and A. Dimitracopoulou. 2006. "Monitoring and Analyzing Group Interactions in Asynchronous Discussions with the DIAS System." *Groupware: Design, implementation, and use* 54–61.
- Brodie, L., and P. Gibbings. 2007. "Developing Problem Based Learning Communities in Virtual Space." in *ConnectED 2007: International Conference on Design Education*.
- Brooke, J. 1996. "SUS-A Quick and Dirty Usability Scale." *Usability evaluation in industry* 189(194):4–7.
- Buck Institute for Education. 2014. "What Is Project Based Learning?" *Buck Institute for Education*. Retrieved (http://bie.org/about/what_pbl).
- Butler, DL, and PH Winne. 1995. "Feedback and Self-Regulated Learning: A Theoretical Synthesis." *Review of educational research* 65(3):245–81.
- Charleer, S., S. Odriozola, and J. Luis. 2014. "LARAe: Learning Analytics Reflection & Awareness Environment." in *ARTEL14: Proceedings of the 4th Workshop on Awareness and Reflection in Technology-Enhanced Learning*.

- Chen, CM. 2009. "Personalized E-Learning System with Self-Regulated Learning Assisted Mechanisms for Promoting Learning Performance." *Expert Systems with Applications* 36(5):8816–29.
- Chu, SKW, CKK Chan, and AFY Tiwari. 2012. "Using Blogs to Support Learning during Internship." *Computers & Education* 58(3):989–1000.
- Clark, G. T., and R. Mulligan. 2011. "Integrating a Traditional Problem Based Learning Course with a Learning Management System." *Dental medicine research* 31(3):221–27.
- Cleary, TJ, and BJ Zimmerman. 2004. "Self-regulation Empowerment Program: A School-based Program to Enhance Self-regulated and Self-motivated Cycles of Student Learning." *Psychology in the Schools* 41(5):537–50.
- Despres, C., and T. Coffinet. 2004. "Reflét, Un Miroir Sur La Formation." *ACTES DU COLLOQUE TICE 2004- «Sessions Recherche »* 19–24.
- Després C., Leroux P. 2002. "Un Outil de Suivi Synchrone Pour La Formation À Distance." pp. 315–20 in *Technologies de l'Information et de la Communication dans les Enseignements d'ing énieurs et dans l'industrie*. Lyon.
- Diagne, F. 2009. "Instrumentation de La Supervision Par La Réutilisation D'indicateurs: Modèles et Architecture." Université Joseph Fourier.
- Dimitrakopoulou, A., A. Petrou, and A. Martinez. 2006. *State of the Art of Interaction Analysis for Metacognitive Support & Diagnosis*.
- Djouad, T., and A. Mille. 2010. "Collaborative Activity Indicators Engineering: Using Modeled Traces in the Context of Technology Enhanced Learning Systems." in *Rapport de recherche RR-LIRIS-2010-014*.
- Donnelly, R., and M. Fitzmaurice. 2005. "Collaborative Project-Based Learning and Problem-Based Learning in Higher Education: A Consideration of Tutor and Student Roles in Learner-Focused Strategies." pp. 87–98 in *Emerging issues in the practice of university learning and teaching*.
- Du, HS, and C. Wagner. 2007. "Learning with Weblogs: Enhancing Cognitive and Social Knowledge Construction." *IEEE Transactions on Professional Communication* 50(1):1–16.
- Durall, E., and T. Toikkanen. 2013. "Feeler: Feel Good and Learn Better." *Third International Workshop on Awareness and Reflection in Technology-Enhanced Learning (ARTEL 2013)* 83–89.

- English, M. C. 2013. "The Role of Newly Prepared PBL Teachers' Motivational Beliefs and Perceptions of School Conditions in Their Project Based Learning Implementation." George Mason University.
- English, M. C., and A. Kitsantas. 2013. "Supporting Student Self-Regulated Learning in Problem-and Project-Based Learning." *Interdisciplinary journal of problem-based learning* 7(2).
- Ferguson, R., S. Buckingham Shum, and R. Deakin Crick. 2011. "EnquiryBlogger: Using Widgets to Support Awareness and Reflection in a PLE Setting." pp. 28–33 in *1st Workshop on Awareness and Reflection in Personal Learning Environments*.
- Fessl, Angela, Gudrun Wesiak, and Granit Luzhnica. 2014. "Application Overlapping User Profiles to Foster Reflective Learning at Work." pp. 51–64 in *4th Workshop on Awareness and Reflection in Technology-Enhanced Learning*.
- Gehlbach, H., and RW Roeser. 2002. "The Middle Way to Motivating Middle School Students by Avoiding False Dichotomies." *Middle School Journal* 33(3):39–46.
- Geier, R., and PC Blumenfeld. 2008. "Standardized Test Outcomes for Students Engaged in Inquiry - based Science Curricula in the Context of Urban Reform." *Journal of Research in Science Teaching* 45(8):922–39.
- Gendron, É. 2010. "Cadre Conceptuel Pour L'élaboration D'indicateurs de Collaboration À Partir Des Traces D'activité." Université Claude Bernard-Lyon I.
- George, S., and P. Leroux. 2001. "Project-Based Learning as a Basis for a CSCL Environment: An Example in Educational Robotics." pp. 269–76 in *First European Conference on Computer-Supported Collaborative Learning (Euro-CSCL 2001)*.
- George, S., and P. Leroux. 2002. "An Approach to Automatic Analysis of Learners' Social Behavior during Computer-Mediated Synchronous Conversations." *Intelligent Tutoring Systems* 630–40.
- Glaser-Zikuda, M., and JAN Fendler. 2011. "Fostering Self-Regulated Learning with Portfolios in Schools and Higher Education." *ORBIS SCHOLAE* 5(2):67–78.
- Govaerts, S., K. Verbert, E. Duval, and A. Pardo. 2012. "The Student Activity Meter for Awareness and Self-Reflection." pp. 869–84 in *In CHI'12 Extended Abstracts on Human Factors in Computing Systems*.
- Graf, S., C. Ives, N. Rahman, and A. Ferri. 2011. "AAT: A Tool for Accessing and Analysing Students' Behaviour Data in Learning Systems." pp. 174–79 in *1st International Conference on Learning Analytics and Knowledge*.

- Gueraud, V., and JM Cagnat. 2004. "Suivi À Distance de Classe Virtuelle Active." pp. 377–83 in *Technologies de l'Information et de la Connaissance dans l'Enseignement Supérieur et de l'Industrie*.
- Gültekin, M. 2005. "The Effect of Project Based Learning on Learning Outcomes in the 5th Grade Social Studies Course in Primary Education." *Educational Sciences: Theory & Practice* 5(2).
- Hadwin, AF, and PH Winne. 2001. "Context Moderates Students' Self-Reports about How They Study." *Journal of Educational Psychology* 93(3):477–87.
- Helic, D., and H. Krottmaier. 2005. "Enabling Project-Based Learning in WBT Systems." *International Journal on E-learning* 4(4):445–61.
- Helm, JH, and LG Katz. 2011. *Young Investigators: The Project Approach in the Early Years*. Teachers College Press.
- Jeremic, Z. 2009. "Semantically-Enabled Project-Based Collaborative Learning of Software Patterns." pp. 569–71 in *Ninth IEEE International Conference on Advanced Learning Technologies*.
- Ji, M. 2012. "Ergonomic Study of Existing Project-Based Learning Management System." pp. 57–62 in *Les quatrièmes rencontres jeunes chercheurs en EIAH (RJC EIAH 2012)*.
- Ji, M., C. Michel, E. Lavoué and S. George. 2013. "Combinaison de Traces D'activités et de Reporting Pour Soutenir Des Processus D'autorégulation." pp. 267–78 in *6ème Conférence Nationale sur les Environnements Informatiques pour l'Apprentissage Humain (EIAH 2013)*.
- Ji, M., C. Michel, É. Lavoué, and S. George. 2014. "DDART, a Dynamic Dashboard for Collection, Analysis and Visualization of Activity and Reporting Traces." pp. 440–45 in *Ninth European Conference on Technology Enhanced Learning (EC-TEL)*.
- Kennedy, GJ. 2005. "Peer-Assessment in Group Projects: Is It Worth It?" pp. 59–65 in *7th Australasian conference on Computing education*.
- Krug, S. 2000. *Don't Make Me Think!: A Common Sense Approach to Web Usability*. New Riders.
- Ku, DT, and CS Chang. 2011. "A Framework of PBL Strategy Integrated in LMS and a Ubiquitous Learning Environment." pp. 722–26 in *Sixth International Conference on Networked Computing and Advanced Information Management (NCM)*.

- Laffey, J., T. Tupper, D. Musser, and J. Wedman. 1998. "A Computer-Mediated Support System for Project-Based Learning." *Educational Technology Research and Development* 46(1):73–86.
- Lamprey, WNL, and AR Fayek. 2012. "Developing a Project Status Dashboard for Construction Project Progress Reporting." *International Journal of Architecture, Engineering and Construction* 1(2):112–20.
- Laperroux, C., P. Leroux, and P. Teutsch. 2006. "TACSI, Un Environnement de Suivi Individuel Dans Une Activité Collective-Première Utilisation Dans Un Contexte D'apprentissage Du FLE." in *Technologies de l'Information et de la Communication dans l'Enseignement Supérieur et l'Entreprise*.
- Loboda, Tomasz D., Julio Guerra, Roya Hosseini, and Peter Brusilovsky. 2014. "Mastery Grids: An Open Source Social Educational Progress Visualization." pp. 235–48 in *Ninth European Conference on Technology Enhanced Learning (EC-TEL 2014)*.
- May, M., S. George, and P. Prévôt. 2011. "TrAVis to Enhance Students' Self-Monitoring in Online Learning Supported by Computer-Mediated Communication Tools." *Computer Information Systems and Industrial Management Applications* 3:623–34.
- Mazza, R., and V. Dimitrova. 2007. "CourseVis: A Graphical Student Monitoring Tool for Supporting Instructors in Web-Based Distance Courses." *International Journal of Human-Computer Studies* 65(2):125–39.
- Mazza, R., and C. Milani. 2004. "Gismo: A Graphical Interactive Student Monitoring Tool for Course Management Systems." pp. 18–19 in *International Conference on Technology Enhanced Learning*.
- Mergendoller, J., T. Markham, J. Ravitz, and J. Larmer. 2013. "Pervasive Management of Project Based Learning: Teachers as Guides and Facilitators." in *Handbook of classroom management: Research, practice, and contemporary issues*, edited by E Emmer, E Sabornie, C. M Evertson, and C. S Weinstein. Routledge.
- Michel, C., and É. Lavoué. 2011. "KM and Web 2.0 Methods for Project-Based Learning." *Multiple Perspectives on Problem Solving and Learning in the Digital Age* 49–63.
- Michel, C., E. Lavoué and L. Pietrac. 2012. "A Dashboard to Regulate Project-Based Learning." *21st Century Learning for 21st Century Skills* 250–63.
- Narciss, S., A. Proske, and H. Koerndle. 2007. "Promoting Self-Regulated Learning in Web-Based Learning Environments." *Computers in Human Behavior* 23(3):1126–44.

- Nicol, DJ, and D. Macfarlane-Dick. 2006. "Formative Assessment and Self-Regulated Learning: A Model and Seven Principles of Good Feedback Practice." *Studies in higher education* 31(2):199–218.
- Nielsen, J. 1994. *Usability Engineering*. Elsevier.
- Norman, GR, and HG Schmidt. 1992. "The Psychological Basis of Problem-Based Learning: A Review of the Evidence." *Academic medicine* 67(9):557–65.
- Paris, SG, and P. Winograd. 1999. "The Role of Self-Regulated Learning in Contextual Teaching: Principals and Practices for Teacher Preparation." *Contextual teaching and learning: Preparing teachers to enhance student success in and beyond school* 219–52.
- Perry, NE, and PH Winne. 2006. "Learning from Learning Kits: gStudy Traces of Students' Self-Regulated Engagements with Computerized Content." *Educational Psychology Review* 18(3):211–28.
- Phielix, C., FJ Prins, and PA Kirschner. 2010. "Group Awareness of Social and Cognitive Behavior in a CSCL Environment." pp. 230–37 in *9th International Conference of the Learning Sciences*.
- Pintrich, PR. 1995. "Understanding Self - regulated Learning." *New directions for teaching and learning* (63):3–12.
- Pintrich, PR. 2000. "The Role of Goal Orientation in Self-Regulated Learning." pp. 451–502 in *Handbook of self-regulation*. Academic Press.
- Pintrich, PR, and A. Zusho. 2007. "Student Motivation and Self-Regulated Learning in the College Classroom." pp. 731–810 in *The scholarship of teaching and learning in higher education: An evidence-based perspective*. Springer Netherlands.
- Rasmussen, NH, M. Bansal, and CY Chen. 2009. *Business Dashboards: A Visual Catalog for Design and Deployment*. John Wiley & Sons.
- Reffay, C., and T. Chanier. 2003. "How Social Network Analysis Can Help to Measure Cohesion in Collaborative Distance-Learning." pp. 343–52 in *Designing for change in networked learning environments*. Springer Netherlands.
- Sánchez, RA, and AD Hueros. 2010. "Motivational Factors That Influence the Acceptance of Moodle Using TAM." *Computers in human behavior* 26(6):1632–40.
- Santos, JL, S. Charleer, and G. Parra. 2013. "Evaluating the Use of Open Badges in an Open Learning Environment." pp. 314–27 in *Scaling up Learning for Sustained Impact*. Springer Berlin Heidelberg.

- Santos, JL, S. Govaerts, K. Verbert, and E. Duval. 2012. "Goal-Oriented Visualizations of Activity Tracking: A Case Study with Engineering Students." pp. 143–52 in *2nd international conference on learning analytics and knowledge*.
- Settoui, LS, and Y. Prie. 2009. "A Trace-Based System for Technology-Enhanced Learning Systems Personalisation." pp. 93–97 in *Ninth IEEE International Conference on Advanced Learning Technologies*.
- Smidts, D. 2003. "Un Outil Audiovisuel Pour Former Des Tuteurs En Pédagogie Active." in *2ième Colloque Questions de pédagogie dans l'enseignement sup érieur*.
- Sulaiman, F. 2013. "Physics Students' Acceptance of Pbl Online in Terms Of Learning Outcomes." *International Journal of Humanities and Social Science Invention* 2(3):50–55.
- Sun, PC, RJ Tsai, G. Finger, YY Chen, and D. Yeh. 2008. "What Drives a Successful E-Learning? An Empirical Investigation of the Critical Factors Influencing Learner Satisfaction." *Computers & Education* 50(4):1183–1202.
- Teutsch, P., JF Bourdet, and O. Gueye. 2004. "Perception de La Situation D'apprentissage Par Le Tuteur En Ligne." in *les Technologies de l'Information et de la Connaissance dans l'Enseignement Sup érieur et dans l'Industrie*.
- Thomas, JW. 2000. *A Review of Research on Project-Based Learning*.
- Verbert, K., E. Duval, and J. Klerkx. 2013. "Learning Analytics Dashboard Applications." *American Behavioral Scientist* 57(10):1500–1509.
- Winne, PH. 2006. "The Learning Kit Project: Advancing Research on Learning as Learners Learn in Everyday Settings." pp. 975–79 in *Sixth International Conference on Advanced Learning Technologies*.
- Winne, PH, AF Hadwin, and C. Gress. 2010. "The Learning Kit Project: Software Tools for Supporting and Researching Regulation of Collaborative Learning." *Computers in Human Behavior* 26(5):787–93.
- Winne, PH, and D. Jamieson-Noel. 2002. "Exploring Students' Calibration of Self Reports about Study Tactics and Achievement." *Contemporary Educational Psychology* 27(4):551–72.
- Winne, PH, and NE Perry. 2000. "Measuring Self-Regulated Learning." pp. 531–66 in *Handbook of self-regulation*. Academic Press.
- Wolters, CA. 2003. "Regulation of Motivation: Evaluating an Underemphasized Aspect of Self-Regulated Learning." *Educational psychologist* 38(4):189–205.

- Zaiane, OR. 2002. "Building a Recommender Agent for E-Learning Systems." pp. 55–59 in *International Conference on Computers in Education*.
- Zhang, H., K. Almeroth, and A. Knight. 2007. "Moodog: Tracking Students' Online Learning Activities." pp. 4415–22 in *World Conference on Educational Multimedia, Hypermedia and Telecommunications*.
- Zimmerman, BJ. 1989. "Models of Self-Regulated Learning and Academic Achievement." pp. 1–25 in *Self-regulated learning and academic achievement*.
- Zimmerman, BJ. 1998. "Developing Self-Fulfilling Cycles of Academic Regulation: An Analysis of Exemplary Instructional Models." in *Self-regulated learning: From teaching to self-reflective practice*. Guilford Publications.
- Zimmerman, BJ. 2002. "Becoming a Self-Regulated Learner: An Overview." *Theory into practice* 41(2):64–70.
- Zimmerman, BJ, S. Bonner, and R. Kovach. 1996. *Developing Self-Regulated Learners: Beyond Achievement to Self-Efficacy*. American Psychological Association.

