

Contextual Inquiry Reflection Tools

Citation for published version (APA):

Santos, J. L., Charleer, S., Krmpotic, D., Georgiev, A., Klerkx, J., Haimala, F., & Specht, M. (2013). *Contextual Inquiry Reflection Tools*.

Document status and date:

Published: 31/03/2013

Document Version:

Peer reviewed version

Document license:

CC BY-NC-SA

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

<https://www.ou.nl/taverne-agreement>

Take down policy

If you believe that this document breaches copyright please contact us at:

pure-support@ou.nl

providing details and we will investigate your claim.

Downloaded from <https://research.ou.nl/> on date: 05 Oct. 2024

Open Universiteit
www.ou.nl



weSPOT

IST (FP7/2007-2013) under grant agreement N° 318499

Deliverable number	<i>D5.3</i>
Deliverable Title	<i>Contextual Inquiry Reflection Tools</i>
Dissemination level	<i>PU</i>
Delivery date	<i>31 March 2013</i>
Status	<i>Draft</i>
Author(s)	<i>José Luis Santos, Sven Charleer, David Krmpotic, Atanas Georgiev, Joris Klerkx, Fotini Haimala</i>
Contributors	<i>José Luis Santos, Sven Charleer, David Krmpotic, Atanas Georgiev, Joris Klerkx, Fotini Haimala, Marcus Specht</i>

The weSPOT Consortium

Number	Partner	Country
1	OPEN UNIVERSITEIT NEDERLAND	Netherlands
2	THE OPEN UNIVERSITY	United Kingdom
3	TECHNISCHE UNIVERSITAET GRAZ	Austria
4	KU LEUVEN	Belgium
5	SOFIISKI UNVERSITET SVETI KLIMENT OHRIDSKI	Bulgaria
6	FRIEDRICH-ALEXANDER-UNIVERSITAT ERLANGEN NURNBERG	Germany
7	FOUNDATION FOR RESEARCH AND TECHNOLOGY HELLAS	Greece
8	LATTANZIO LEARNING SPA	Italy
9	IPAK INSTITUT ZA SIMBOLNO ANALIZO IN RAZVOJ INFORMACIJSKIH TEHNOLOGIJ VENENJE ZAVOD	Slovenia

Document Control

Coordinating Editor: *José Luis Santos*

E-mail: joseluis.santos@cs.kuleuven.be

Amendment History

Version	Date	Author/Editor	Reviewers	Description/Comments
1	28/02/2014	José Luis Santos		Structure
2	10/03/2013	José Luis Santos		First Draft -Contribution
3	11/03/2014	Sven Charleer		First Draft -Contribution
4	12/03/2014	David Krmpotic		First Draft -Contribution
5	15/03/2014	Atanas Georgiev		First Draft -Contribution
6	17/03/2014	José Luis Santos		Finishing First Draft
7	18/03/2014		Alexander Mikroyannidis	
8	24/03/2014		Elisabetta Parodi	
9	25/03/2014	José Luis Santos		Final Version

Executive Abstract

The aim of D5.3 is to present the weSPOT Contextual Inquiry Reflection Tools. This deliverable presents a set of tools for personal reflection and deliver a set of component and software applications for personal reflection support. Being this deliverable meant as a prototype it is mainly constituted by the software modules themselves. The related code is already made public in repositories such as GitHub (URLs can be found throughout the document). The present document shortly presents such tools. Scenario 3 presented in TRAC¹ is also updated as part of the deliverable.

¹ <http://trac.wespot.net/wiki/UserScenarioPage>

Table of Contents

Executive Abstract.....	5
Table of Contents	6
1. Introduction	7
2. Context Inquiry Reflection tools in weSPOT	9
3. Context Inquiry Reflection tools in weSPOT	10
3.1. Design based research.....	10
3.2. Badges widget	10
3.3. Learning Analytics Reflection & Awareness environment.....	11
3.3.1 Widget.....	11
3.3.2 Dashboard	13
3.3.3. Evaluating LARAE with the pedagogical group.....	15
4. Tracking data	16
4.1. Tracking Elgg/Workflow Inquiry Engine data.....	16
4.1.1 What is Elgg	16
4.1.2 Plugin architecture	16
4.1.3 Our plugin for tracking Elgg user actions.....	16
4.1.4 Advantages of our approach	17
4.1.5 Possible improvements.....	17
5. Storage layer	19
5.1. Refactoring of the services data store	19
5.1.1. Pushing data	19
5.2.2. Pulling data.....	20
5.2. Badge API.....	20
5.2.1 Management of badges.....	20
5.2.2 Management of awarded badges	21
5.3. Hadoop.....	22
6. Conclusions and future work.....	24
References.....	25

1. Introduction

One of the major innovative elements of the weSPOT pedagogical approach is the key role of students' reflection in the inquiry process. As it has been reported in D2.3.1², the weSPOT inquiry based learning model places reflection at the centre of each inquiry phase, sees it as an integrated process throughout the inquiry activity and not as an independent phase that comes at the end of the process. This is mainly due to the fact that reflection is vital at every stage of the inquiry, even at the very beginning when students need to develop a question or a hypothesis. Students need to reflect upon the question, and evaluate it before they decide to proceed. They also need to reflect while deciding what kind of data they need to collect, how to proceed to data analysis, how to communicate their results. To highlight the importance of reflection at each stage of the inquiry process and not only at the end is essential for successful and effective scientific inquiries. Indeed successful scientific investigations require from the investigator certain mental attitudes such as, curiosity, scepticism, and objectivity ([Hunt and Colander 2010](#)), which are more likely to be developed when students reflect at every step of their inquiries.

In the view of the key role of the reflection in the process of students' inquiries, this deliverable describes our ongoing work regarding T5.3 within WP5. T5.3 is described in the DoW as:

In this task the inquiry analytics toolkit will be extended social and didactical contexts to stimulate reflection about personal achievements in this framing. One aspect is to contextualize the awareness data, for instance by relating them to those of peer learners, or to the budget or goal of resource use defined by the teacher. Another aspect is support for realistic goal setting by the user, in order to create feedback loops, where the learnscape can help to steer future activities by the learner, and enable self-improvement. A final aspect is using the ambient nature of displays to provide unobtrusive subtle hints in learning settings that can help learners to carry out relevant activities.

However, the main outcomes of this task are the component themselves. The URLs to the source code are included along the description of the tools in this deliverable. The progress on this task can be checked on TRAC where the Scenario 3: "The class looks at collected data and discusses the collected data" has been updated³.

This deliverable also reports on the evolution from D5.2 Inquiry Reflection tools. StepUp!, Navi and the FCA tool, more concretely this deliverable focuses on what are the decisions made to reinforce the integration of StepUp! and Navi with the project. Trackers and Data Store remain the same concept as described in D5.2.

Task 5.3 runs from M9 to M30, focusing the higher workload from M19 to M27. Therefore, the work presented in this deliverable is not definitive but work-in-progress.

The work described in this deliverable will be presented into two publications submitted to EC-TEL by the 13th of April.

The work done described in this deliverable relates on two main motivations:

- Motivation, while often overlooked, is perhaps the single most important factor in learning. Also in our earlier work, students reported motivation as one of their main learning issues. New strategies to raise learner motivation have been discussed in recent years. . Exciting new research is adopting badge systems to certify skills and abilities ([Young,2012](#)) such as Mozilla Open Badges that allows students to share their achievements/badges on different social networks such as blogs, LinkedIn and Facebook([Goligoski, 2012](#)). The use of badges has been proven to positively increase learner motivation ([Domínguez et al, 2013](#)). In addition, badges are a representation of how students carried out the activities, which in turn can affect positively the learner motivation ([Michel et al., 2012](#)). Earlier research in this project confirmed positively aspects on this matter ([Santos et al., 2013](#)).

² <http://portal.ou.nl/documents/7822028/3b2c3110-98bf-4686-b2bd-5756fcab93f1>

³ <http://trac.wespot.net/wiki/UserScenarioPage>

- Increasing motivation, autonomy, effectiveness, and efficiency of learners and teachers is an important driver for learning analytics research (Buckingham et al., 2012). In our work, we focus on the microlevel of individual learners and teachers as well as on learning communities that form around courses, mostly in an open learning context (Govaerts et al., 2011). We do not consider the mesolevel of the organization (school, university, training department) or the macrolevel of society at large. We deploy information visualization techniques in dashboard applications for learners, on both mobile devices and desktop as well as tabletop devices. Adopting a “modest computing” approach that tries to empower people, rather than automating decisions on their behalf (Dillenbourg et al., 2011), we focus on approaches that rely on visualization of these traces to assist users (Verbert et al., 2013). Previous evaluations (Charleer et al., 2013) pointed out that the connection of the analytics with the content generated could enhance the reflection process contextualizing the analytics with the real content triggering deeper reflections. The dashboard will play an important role in the phases Interpretation and communication described in D2.3.1⁴. Hands on material (see poster attachment) will be provided supporting this role of the dashboard within the project.

Therefore, this deliverable focuses on the components highlighted in green of the general architecture in Figure 1. The structure of the deliverable is divided in the following three layers:

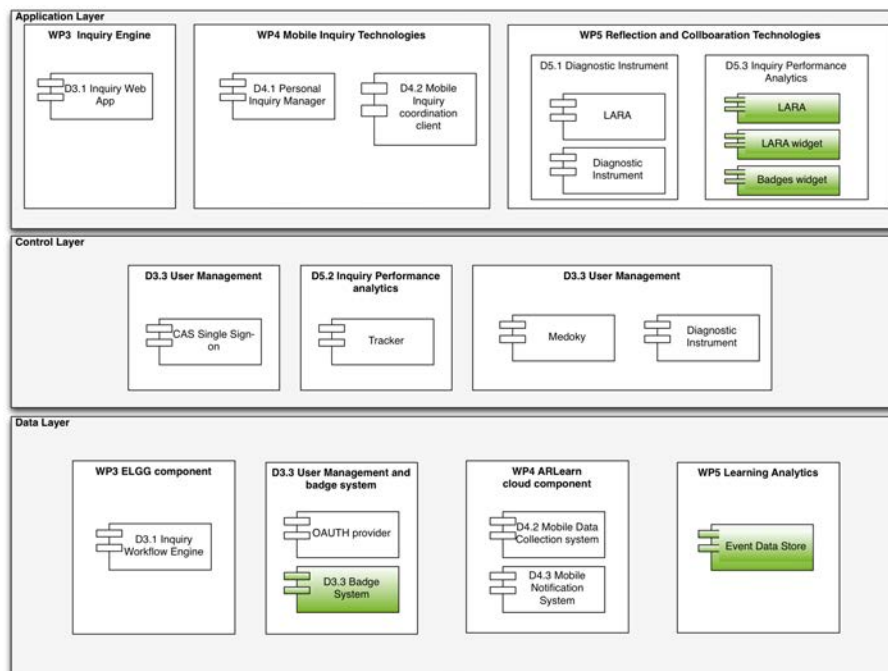


Figure 1 weSPOT architecture

- Application layer (Section 3) :
 - The research methodology (Section 3.1)
 - A Badges Widget (Section 3.2) enables teachers to create, award and visualize the awarded widgets in the class.
 - A the widgetized dashboard version can be deployed in a space of the Inquiry Workflow Engine; a dashboard that displays information about the amount of student activity enabling social comparison; the early results of the evaluation with the pedagogic group (Section 3.3).
- Control layer:
 - The tracked data (Section 4) has been improved tracking widgets interactions.

⁴ <http://portal.ou.nl/documents/7822028/3b2c3110-98bf-4686-b2bd-5756fcab93f1>
weSPOT Project - IST (FP7/2007-2013) under grant agreement N° 318499

- Data layer:
 - We have splitted the badge management from the user data management or data store (Section 5.1).
 - We have refactored the services from the badge API and the Data Store API becoming more RESTful compliant (Section 5.2).
 - We deployed Hadoop technology in the Data Store to ensure scalability (Section 5.3).

Finally, we present our conclusions and remaining future work in Section 6.

2. Context Inquiry Reflection tools in weSPOT

The architecture and how the weSPOT components are connected is described in Figure 2.

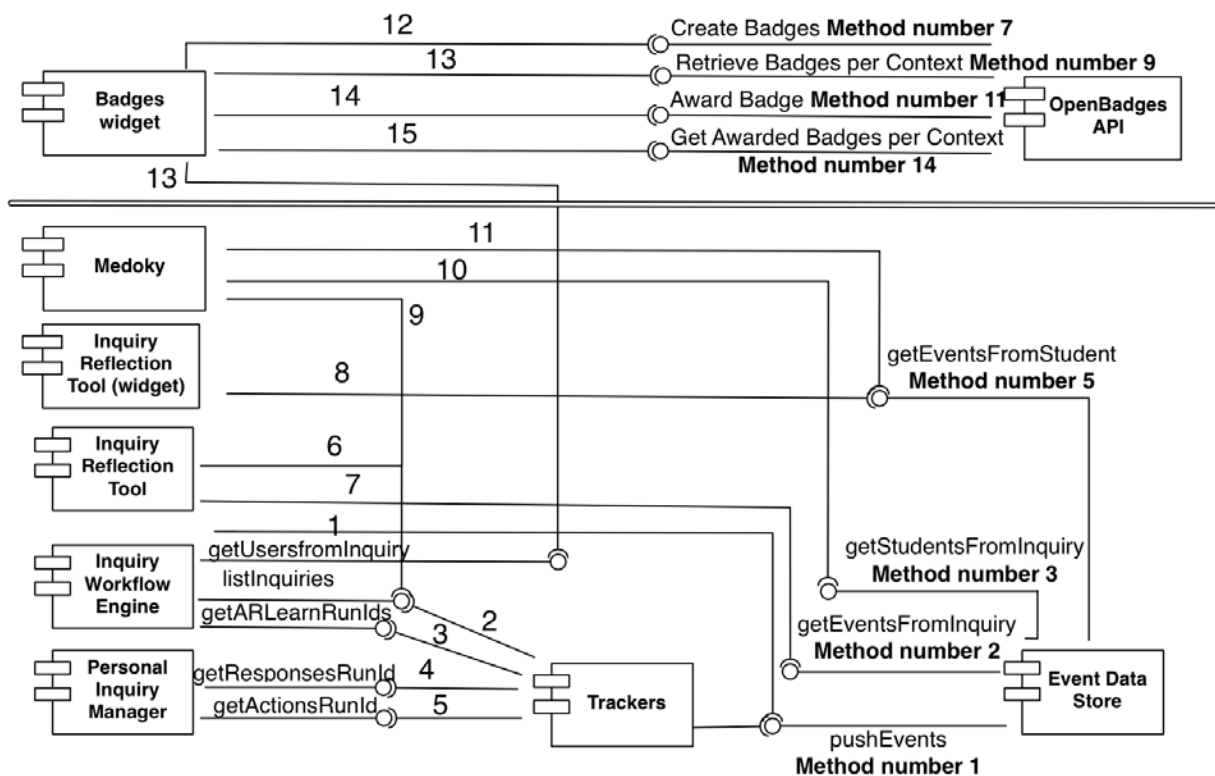


Figure 2 Architecture Diagram. Method numbers correspond to the methods described in Sections 5.1 and 5.2

The architecture of weSPOT regarding the learning Analytics Components is as follows:

- The Inquiry Workflow Engine is a Personalized Learning Environment (PLE) based on Elgg technology. Widgets are deployed on the PLE to perform the different activities in the inquiry. Every widget generates their own events that are pushed to the data store (label 1 at figure 2)
- The Personal Inquiry Manager is the mobile data collector client. It exposes the information via an API that a tracker uses to pull and to store the information in the data store. The Personal Inquiry Manager relies on ARLearn technology in order to collect data such as pictures, videos and text. The sequence is described at Figure 2 (label 2). Trackers get the list of inquiries/courses from the Inquiry Workflow Engine (label 3). For each inquiry, the Trackers get the identifiers of the ARLearn data collection tasks (label 4 & 5). Based on the identifiers, the Trackers pull the data from ARLearn and push it to the Data Store.

- The Inquiry Reflection Tool visualizes the activity and the data from an inquiry. The sequence is defined at Figure 2 (label 6). It gets the list of inquiries from the Inquiry Workflow Engine and (label 7). The Inquiry Reflection Tool gets all the events related to an inquiry. The dashboard visualizes the data of all the students together. Therefore you can see at a glance all the activity.
- The widgetized version of the Inquiry Reflection Tool uses the inquiry id and the student identifier that is logged in at that moment to access and display all the events related to the user (label 8 at Figure 2)
- Medoky is also a consumer of data. The sequence is described at Figure 2: it gets the list of inquiries from the Inquiry Workflow Engine (label 9), it gets all the students from an inquiry (label 10) and then gets all the events related to the student (label 11).
- The Badges Widget connects to the OpenBadges API. The widget has three functionalities described at Figure 2. To create badges is the first functionality (label 12). The second functionality is awarding badges, this one requires two steps: gathering users enrolled in the inquiry and collecting all the available badges in this inquiry (see labels 13), and awarding a badge to a student using the identifiers previously collected (label 14). The third functionality is providing the overview of the badges awarded in the inquiry (label 15).

3. Context Inquiry Reflection tools in weSPOT

3.1. Design based research

The design-based research methodology has been applied to conduct this research. This methodology relies on rapid prototyping to evaluate ideas in frequent short iteration cycles ([Orril et al., 1998](#)). The approach enables to collect both qualitative and quantitative evaluation data during the whole software design process ([Wang et al., 2005](#)).

3.2. Badges widget

The widget prototype (see Figure 3) is an early version that requires clearly further improvements. However, it is the first approach to experiment with the manual badges possibilities. The top-left screen at Figure 3 is the menu that enables the access to the different functionalities. The widget consists of three main functionalities: 1) Create a badge (top-right at Figure 3), 2) Award a badge to a student (bottom-left screen at Figure 3) and 3) List awarded badges (bottom-right screen at Figure 3).

Besides the prototype, T3.3 is defining a set of badges that will be automatically awarded based on the tracked user data defined in Section 3.3.

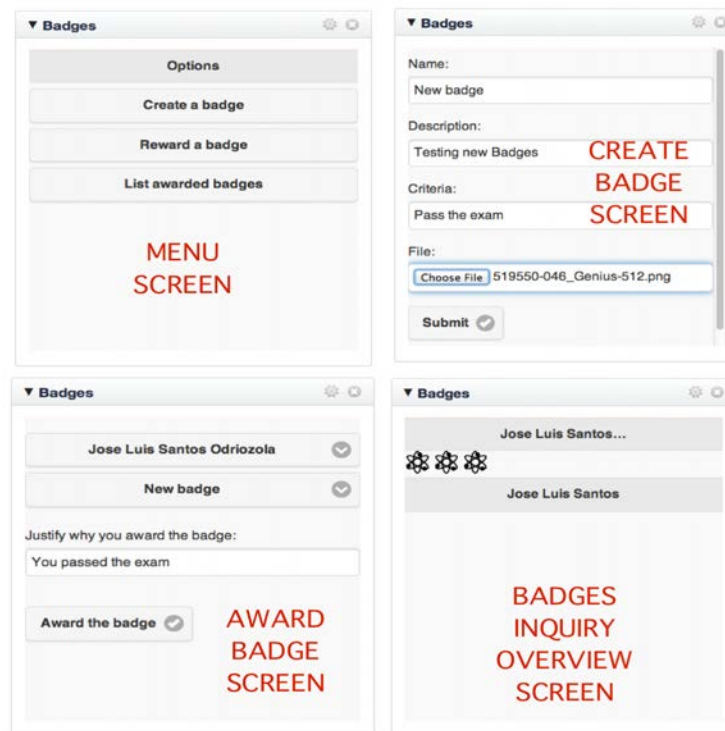


Figure 3 weSPOT badge prototype

3.3. Learning Analytics Reflection & Awareness environment

The Learning Analytics Reflection & Awareness environment (LARAe)⁵ presents teachers and students with visualizations of learner traces. These learner traces are generated by learner activities in the Workflow Inquiry Engine and the Personal Inquiry Manager. Access to this data is provided by the data collection layer as explained in Section 4. LARAe evolved from previous D5.2 tools such as StepUp! and Navi, focussing on abstraction of learner traces and providing access to the valuable data behind the traces helping deeper reflection (Charleer et al., 2013). LARAe is available both through an ELGG widget for a quick personal overview of activities, and through a web dashboard, providing an overview of all activities and generated content in the ILB environment. Both options are described below.

3.3.1 Widget

Through Elgg, the user can access the Learning Analytics widget which presents the user with a quick overview of his/her activities. Figure 4 shows a snapshot of the state of activities of a particular user.

⁵ <https://github.com/svencharleer/LARAe.weSPOT>

Batteries discovery Edit inquiry Invite users

Description:
In this inquiry we will investigate the everyday use of batteries, as well as their impact on the environment.

Tags:
battery, environment

Owner: Jose Luis Santos
Inquiry members: 27
Open membership

Create the questions and hypothesis Plan the method Collect the data Analyse the data Discuss the findings Communicate the results

Add inquiry components

Learning Analytics and Reflection Environment

Question / Hypothesis
Operationalisation
Data Collection
Data Analysis
Interpretation
Communication

Hypotheses

Batteries are really bad for the environment
Last updated 59 days ago by Robin De Croon Comments (1)

Batteries contain heavy metals such as mercury, lead, cadmium, and nickel, which can contaminate the environment when batteries are improperly disposed of. When incinerated, certain metals might be released into the air or can concentrate in the ash...

Larger batteries provide more power!
Last updated 59 days ago by David Krmpotic Comments (7)

More hypotheses

Figure 4 LARAe widget

The widget presents the user with the six phases of the IBL model and the actions related to the current inquiry. Every action the user performs in the inquiry as described in 4.2 is displayed as a circle and positioned chronologically within the phase the action belongs to.

The user instantly sees how active he/she was in each phase and which phases could require more attention. As this data might not be sufficient without the proper context, e.g. the actual value of each activity and a way of comparison with peers, the user can click on any event or the link at the bottom to access the LARAe dashboard.

3.3.2 Dashboard

The screenshot shows the weSPOT interface. At the top left is the logo 'weSPOT Learning Analytics Reflection & Awareness environment' and the user name 'Sven Charleer'. Below the header, there are two inquiry cards. The first card is titled 'Batteries discovery' and contains sections for 'Context', 'Problem', 'Participants', and 'Inquiry based teaching team'. The 'Context' section describes biodiversity in gardens. The 'Problem' section quotes Helen Bostok (RHS) about maximizing garden potential. The 'Participants' section lists several international educators. The second card is titled 'Microclimates' and provides a welcome message for a Year 9 geography fieldwork investigation.

Figure 5 LARAE inquiry overview

The dashboard first presents the user with the list of inquiries he/she has participated in (Fig. 5). The list displays the inquiry names and their description. Clicking on an inquiry will present the LARAE dashboard as depicted in Figure 6.

The dashboard gives an overview of all users' activities in the Inquiry Workflow Engine and PIM's environment. Every column represents a phase of the IBL model while every row represents a user that has participated in the inquiry. The active user is displayed in the top row.

The screenshot shows the LARAE dashboard for the 'Batteries discovery' inquiry. It features a grid where rows represent participants and columns represent phases of the Inquiry-Based Learning (IBL) model: Question/Hypothesis, Operationalisation, Data Collection, Data Analysis, Interpretation, and Communication. Each cell in the grid contains one or more colored circles. Dark circles indicate completed activities, while light circles indicate activities performed on the current day. The top row, representing the active user 'Sven Charleer', shows activity in the 'Operationalisation' phase (one light blue circle) and the 'Communication' phase (multiple dark purple circles). To the right of the grid is an 'Event Data' section containing a photograph of two AA batteries, one of which is an Energizer battery.

Figure 6 LARAE dashboard

As with the widget, each circle matches an activity performed by the participants. Every color matches a phase and every phase color is available in a dark and light color (Figure 7). Light colored circles represent events that happened today, giving an indication on what events are new. Presenting the data in this way, the user can instantly compare activities across participants and discover new events.

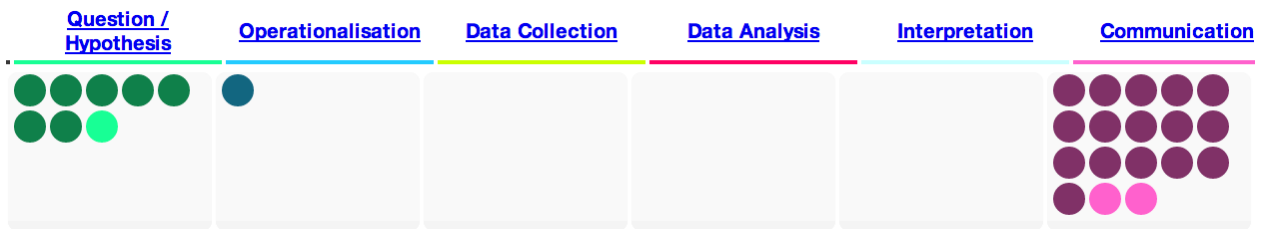


Figure 7 Colored events

Activities are registered for different widgets. Therefore the dashboard allows filtering per widget in each phase. Clicking on the phase name (Figure 8) displays a dropdown containing the widget names. Clicking a widget name will highlight its related activities by rendering them larger and the unrelated activities smaller.

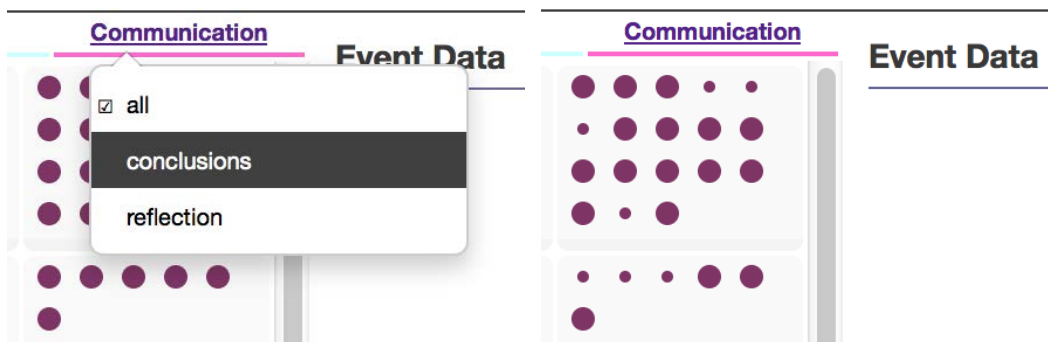


Figure 8 Filtering

An important feature of the dashboard is the access to the content that drives the learning analytics. From this overview, teacher and student can both access the user-generated data created in the Workflow Inquiry Engine and PIM systems. By clicking on an activity, the dashboard displays the Workflow Inquiry Engine data stored in the learning traces, as defined in Section 4. For PIM’s multimedia data, such as videos, images and audio, the dashboard connects to the PIM system and embeds the content into the dashboard (Figure 9).

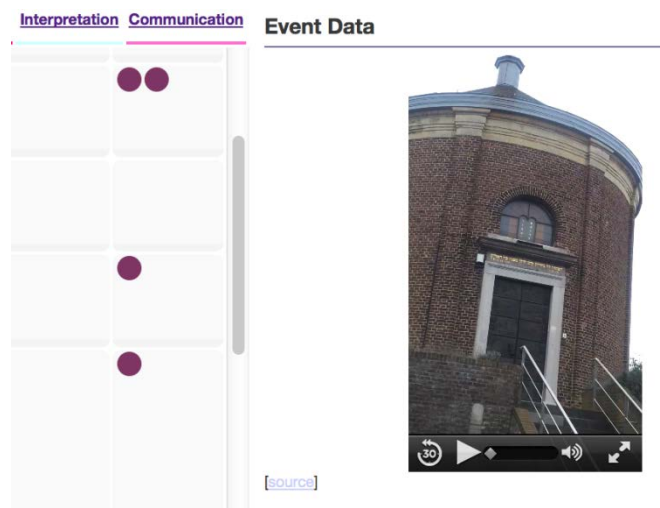


Figure 9 ARLearn multimedia data

The dashboard connects to Workflow Inquiry Engine, the datastore and PIM. Workflow Inquiry Engine is accessed to retrieve the inquiries of the user through a RESTful service. The datastore provides all the learning traces, Workflow Inquiry Engine activity content and links to PIM’s multimedia data and is also accessed through its RESTful service. PIM data is displayed through HTML5 and the PIM web service.

The dashboard is created using Node.js as backend service, HTML5 and D3.js for its frontend visualisations.

3.3.3. Evaluating LARAE with the pedagogical group

We have evaluated the dashboard through a semi-structured interview collecting qualitative data and quantitative data through a questionnaire regarding functionalities plus SUS-questionnaire to evaluate usability and perceived usefulness⁶. We evaluated with 10 people, of which 5 people of the pedagogical partners of weSPOT and 5 technical people who also have pedagogical tasks as part of their daily work.

Everyone agreed that this dashboard provides the user with a good overview of the activities, however 2 users found the dashboard to be more teacher focussed and less useful for students. All the participants also agreed that showing the actual content of each event is extremely useful. Teachers can keep track of the actual activities of students, while students can easily access artifacts generated by their peers which can have a positive impact on their own activities.

One user did express privacy concerns regarding the data shown on the dashboard. Even though the data is always accessible to all users of the Workflow Inquiry Engine system, this more direct overview where no effort is required to access every user's data and compare their activity convinced one participant that it should be limited to teacher use only.

The dashboard leaves room for many improvements. Through these evaluations we have found what would make most sense to prioritize from a pedagogical point of view. Currently, the dashboard visualises all actions even though some actions might not be relevant or have a lesser priority to others. An action like "rating of a question" should probably be weighted differently from "gathering of an artifact". Therefore, some actions could be left out from the visualisation, while others can be used to enrich the visualisation in different ways e.g. the amount of comments on a question can affect the size of that questions' activity circle. Feedback from teachers in the form of comments, ratings or badges could be displayed as checkmarks on user's activities. This all adds to visualising the quality of activities, which 4 participants deemed very important.

The relationship between activities is lost to the user, whether they are using the Inquiry Workflow Engine or the dashboard. While comments on activities are easy to find, understanding what data collection is related to what hypothesis i.e. cross phase activities, becomes harder. The use of tags could help solve this issue. Although it requires an extra effort from both teacher and students, all participants agreed that it would greatly improve the learning environment. One participant even suggested to make the use of tags mandatory.

The existence of tags in the learning traces can help us improve the visualisations and help the user understand the data better. Figure 10 shows a prototype we have deployed in a university course based on this idea: clicking on an activity will highlight related events through the entire visualisation. On the right all the content of these events is displayed in chronological order.

⁶ <http://goo.gl/CeWdsA>

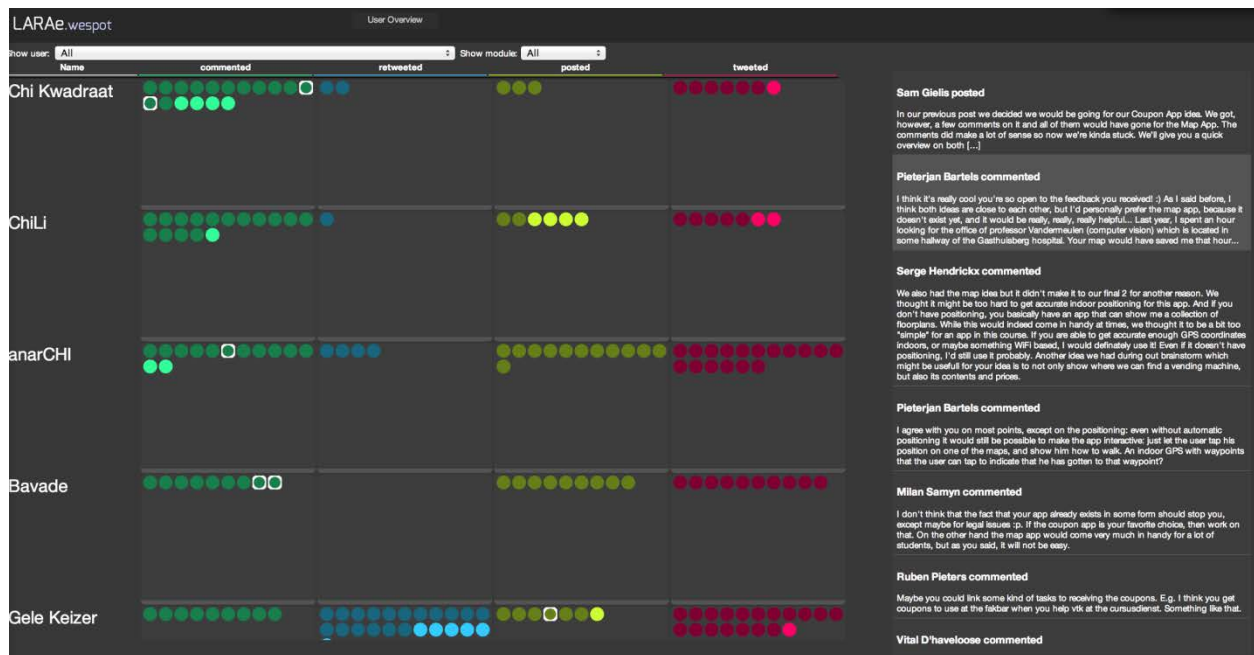


Figure 10 Prototype showing linked data

4. Tracking data

This Section explains how the Inquiry Workflow Engine has been modified in order to push the interaction data generated by users interacting with the system. The Personal Inquiry Manager process remains the same as described in D5.2. The events are listed in Section 4.2.

4.1. Tracking Elgg/Workflow Inquiry Engine data

4.1.1 What is Elgg

Elgg is open-source platform that in essence provides social networking functionality. It can be used as a PLE (Personal Learning Environment) because it supports blogs, file repositories, tagging, user profiles, commenting, communities and more.

Elgg's architecture is modular and so the platform can be customized to one's needs - functionally and visually. Various components can be switched on or off and there are many additional plugins available from the community, for example chat and calendar among others.

Software is translated in multiple languages, which is a big plus.

4.1.2 Plugin architecture

Plugins for Elgg are developed in a standardized way by using main framework functions, defining plugin views and listening to or triggering events. It is easy to extend the platform in this way and provide functionality that was originally not there.

4.1.3 Our plugin for tracking Elgg user actions

We created a plugin with code to intercept different events across the Elgg platform. Our plugin does not provide any visual elements (views).

Events in Elgg are intercepted by registering event handlers like this:

```
elgg_register_event_handler('create', 'object', 'create_object');
```

On each object creation the “create_object” function is called to which the Event, Type and Object are passed. In the handler function we can deduct from which plugin the event was fired by checking the objects’s subtype.

We used about six event handlers - for creating objects, for updating objects, for creating river items, for creating answers, for ratings, for creating annotations and for deleting annotations.

The last one we actually fire ourselves from each plugin on the delete action from the UI - this is the only case where other plugins’ code had to be updated. We couldn’t just handle normal object delete events because when a parent object with a lot of comments was deleted, the handler would be called for each deleted child object.

In each handler we set the data that has to be sent to Data Store Service service to an array and pass that to a “post_data” function. This function uses httpful library’s Request::post function to send json with supplied data to the Data Store Service service.

Because of the bugs in Elgg core when some events are fired multiple times instead of just once, we are checking data and if it’s the same as some other request in last five seconds, we ignore it.

4.1.4 Advantages of our approach

By creating a plugin and responding to framework events, we were able to contain most of the code for tracking in one package. Advantage is that the code can be understood and maintained in central location and if the platform upgrades to a newer version, our code does not get replaced and lost. Another advantage is that the plugin can be simply turned off and the tracking is disabled.

4.1.5 Possible improvements

At the moment on each user action we trigger a request to the Data Store Service analytics server. In case there are a lot of users on the Elgg system and Data Store Service service gets overloaded with requests, it could slow down the experience for the users or maybe even hinder the usability in extreme cases (when Data Store Service would take really long time to process each request). An improvement would be to accumulate data to be sent to Data Store Service in a special queue and process that queue asynchronously. We plan to implement this based on stress testing and if there turns out to be a big difference in final capabilities of Elgg and Data Store Service server(s).

4.2. Data table report

The list of events that are currently tracked in wespot are described in table 1:

Table 1 Tracked events

Phases	Widget/tool	Action	Data
Phase1	Hypothesis	Create	title/description/tags
		Edit	title/description/tags
		Like	id hypothesis

		Rated	
		delete	id hypothesis
		commented	content of the comment and id of the hypothesis
		delete_comment	
	Questions	Create	title/description/tags
		Edit	title/description/tags
		Like	id question
		Rated	
		delete	id question
		reply	content of the reply and id of the hypothesis
		delete_reply	
	Mindmap	create	id, title, description
		edit	id, title, description
		delete	id mindmap
Phase 2	Notes	<i>Same like hypothesis</i>	
	Pages	<i>Same like hypothesis</i>	
	Mindmap		
Phase 3	Data Collection widget	Create	id/title/description/tags/type
		Edit	id/title/description/tags/type
	ARLearn	startrun	runId
		answer_given	url of the generated resource
	Mindmap	<i>Same like in phase 1</i>	
Phase 4	files	upload	id, title, description

	mindmap	<i>Same like in phase 1</i>	
Phase 5	Inquiry_discussion	<i>Same like hypothesis</i>	
	Questions	<i>like phase 1</i>	
Phase 6	Conclusions	<i>Same like hypothesis</i>	
	Reflections	<i>Same like hypothesis</i>	
	Mindmap	<i>Same like in phase 1</i>	
	Pinboard	<i>TBD</i>	

5. Storage layer

5.1. Refactoring of the services data store

The Data Store⁷ is the glue between all the previously described applications as it stores the information streams. The services to push and search events have been refactored in order to be more RESTFUL compliant. The API⁸ is defined as follows:

5.1.1. Pushing data

The Data Store enables third-party applications to store the data via two http calls described in Table 2 (Number links with Figure 2).

Table 2 Pushing services

Number	1	2
Method	Post	Post
Path	/event	/events
Accept	JSON following event specification	Array of JSONs following event specification
Produces	JSON containing error if produced with the format	JSON containing error if produced with the format
Exceptions	Authentication exception	Authentication exception

⁷ <https://github.com/jlsantoso/stepup/tree/chicourse/StepUpDBInterface>

⁸ <http://wespot.net/en/apis>

5.2.2. Pulling data

All the calls enable to filter the data by time (from, to) and pagination(pag) using GET query parameters. The pagination enables to go over the limit of the results. The result limit is set to 1000 events but can be configured.

The services assume that the logic of the course such as structure of the course and roles is contained in other services such as Learning Management System that expose the required information.

These services have been implemented based on requirements of third-party tools and they are currently used. These services can be easily extended as we will explain in the next chapter.

The services are described in Table 3 (N links with Figure 1).

Table 3 Retrieving services

N	Method	Path	Params	Produces	Exception
3	GET	/context/{context}/users	context: contextId	JSON with students ids	Authentication
4	GET	/context/{context}/events	context: contextId	JSON containing events	Authentication
5	GET	/context/{context}/context/{user}/events	context: contextId user: userId	JSON containing events	Authentication
6	GET	/context/{context}/verb/{verb}/events	course: courseId verb: verbId	JSON containing events	Authentication

5.2. Badge API

We have splitted the Badge API^{9,10} from the Data Store as it was described in D5.2. The main reason was to have an independent badge management system that supports badge creation and award.

5.2.1 Management of badges

Table 4 Services to manage badges

N	Method	Path	Params	Produces	Exception
7	POST	/badges	- context (mandatory) Inquiry identifier - version (mandatory) Badge version number - name (mandatory) Badge name - image (mandatory) URL to the badge image - criteria (mandatory) Explanation of the criteria	JSON with badge identifier	Authentication

⁹ <https://github.com/jlsantoso/stepup/tree/chiccourse/OpenBadgesAPI>

¹⁰ <http://wespot.net/en/apis>

			that leads to awarding the badge		
8	GET	/badges/badgeldentifier/{badg eld}	badgeld: badgeld	JSON containing the searched badge	Authentication
9	GET	/badges/context/{context}	context: contextId	JSON Array containing all the badges created under the defined context	Authentication

5.2.2 Management of awarded badges

Table 5 Services to manage awarded badges

N	Method	Path	Params	Produces	Exception
11	POST	/badges/badgeld/{badgeld}/aw ard	- badgeld (mandatory) Identifier of the badge that will be awarded -recepient (mandatory) user identifier from the user that will be warded with the badge with id=badgeld -evidence (mandatory) Ideally, this character string represents the event ids that explain why the badge has been awarded.	JSON with awardedbadge identifier	Authentication
12	GET	/badges/badgeld/{badgeld}/aw arded	- badgeld (mandatory) The badge for that can be awarded to users from (optional) Return only badges that were awarded after this POSIX timestamp	JSON containing the requested badge	Authentication
13	GET	/badges/user/{userId}/awarded	- userId (mandatory) The user for which awarded badges are returned - from (optional) Return only badges that were awarded after this POSIX timestamp	JSON Array containing all the badges that the user has been awarded with	Authentication

			- resumptionToken (optional) Token to retrieve the next badge of results. Use this token in combination with the same parameters as specified in the original request.		
14	GET	/badges/context/{context}/awarded	- context (mandatory) The context (Inquiry identifier) for which awarded badges are returned -from (optional) Return only badges that were awarded after this POSIX timestamp - resumptionToken (optional) Token to retrieve the next badge of results. Use this token in combination with the same parameters as specified in the original request.	JSON Array containing all the badges that has been awarded in the requested context	Authentication

5.3. Hadoop

This Section describes the ongoing progress of the suggested enhancements of the data store infrastructure with Hadoop as described in deliverable D5.2.

The proposed solution consists of the following components:

- Hadoop Distributed File System (HDFS) Sqoop
- Apache Hive
- Apache Sqoop
- Hadoop FDW (Foreign Data Wrapper)

In order to reduce the initial effort for setting up and configuring the whole environment we have decided to use BigSQL™ (<http://www.bigsql.org/se/>) which provides a preconfigured environment with most of the above components.

The integration between Data Store and Hadoop environment consists of several steps described below.

5.3.1 Importing existing data from PostgreSQL into Hadoop

First we need to dump the event data in a suitable format in order to be properly used in Hadoop. For this task we have additionally installed and configured Sqoop.

Next we have created a Sqoop job that imports the data from the PostgreSQL event table into Hive table. The script for the Sqoop job is shown below.

```
sqoop job --create eventjob -- import \  
--connect jdbc:postgresql://localhost:5432/stepup?charSet=UNICODE \  
--hive-import \  
--hive-table event \  

```

```
--query 'SELECT event_id,username,verb, \
        starttime,endtime,object,target,location,context,timestamp \
        FROM event WHERE $CONDITIONS' \
--hive-drop-import-delims \
--null-string '\\N' --null-non-string '\\N' \
--input-null-string '\\N' --input-null-non-string '\\N' \
--username postgres --password password \
--incremental append --check-column event_id --split-by event_id \
--target-dir /user/bigsql/event
```

The job performs incremental import by examining the value of the event_id column to determine which rows to import.

Using Linux crontab the job is scheduled to be executed periodically and importing only fresh data.

5.3. 2 Accessing Hadoop from within PostgreSQL

The Hadoop FDW (Foreign Data Wrapper) helps to leverage the power of Hadoop from within PostgreSQL giving native/optimized access to perform low latency analytics on the events table data stored in Hadoop using MapReduce constructs.

We have created a foreign table and thus allowing querying this "foreign table" as if it is inside of PostgreSQL and even allowing joining native PostgreSQL tables to it.

The psql script for the foreign table definition is shown below.

```
CREATE SERVER hadoop_server FOREIGN DATA WRAPPER hadoop_fdw
  OPTIONS (address '127.0.0.1', port '10000');
CREATE USER MAPPING FOR PUBLIC SERVER hadoop_server;
DROP SCHEMA IF EXISTS hive CASCADE;
CREATE SCHEMA hive;
DROP FOREIGN TABLE IF EXISTS hive.event;
CREATE FOREIGN TABLE hive.event (
  event_id INT NOT NULL,
  username character varying(100) NOT NULL,
  verb character varying(100) NOT NULL,
  starttime TIMESTAMP,
  endtime TIMESTAMP,
  object character varying(300) NOT NULL,
  target character varying(300),
  location text,
  context character varying(100),
  "timestamp" TIMESTAMP)
SERVER hadoop_server
OPTIONS (
  table 'event'
);
```

The deployed Hadoop environment can be used for enhancement of the core analytics and reporting infrastructure by providing high availability and flexible scalability for rapid data growth.

6. Conclusions and future work

This deliverable describes the evolution of the reflection tools in weSPOT. Our efforts focused on improving the integration of our tools in the Inquiry Workflow Engine. To this end, we have developed two new widgets that enable the dashboard deployment and the creation and badges management in an inquiry.

We have deployed a new version of the dashboard and evaluated with the pedagogical group getting very positive response about the changes.

Regarding the backend, we have standardised the services and we have separated the badges management from the user data management that relied on the data store. Finally, we have worked on potential scalability issues that could appear due to the data growth.

Besides the weSPOT project, we are evaluating the same dashboard approach in order to get some feedback and proceed to some changes. On the other hand, pilots are starting to deploy evaluations from which we expect to gather feedback and proceed with further changes.

References

- Charleer, S., Klerkx, J., Santos, J.L., Duval, E., (2013) Improving awareness and reflection through collaborative, interactive visualizations of badges. Proceedings at ARTEL'13 workshop. CEUR-WS. pp 69-81 (Attached to the deliverable, but this publication was already submitted with D5.2)
- Buckingham Shum, S., Gašević, D., & Ferguson, R. (Eds.). (2012). Proceedings of 2nd International Conference on Learning Analytics and Knowledge, LAK12. ACM.
- Domínguez, A., Saenz-de Navarrete, J., de Marcos, L., Fernández-Sanz, L., Pagés, C., Martínez-Herráiz, J.J. (2013) Gamifying learning experiences: Practical implications and outcomes. *Computers & Education* 63, 380–392
- Dillenbourg, P., Zufferey, G., Alavi, H., Jermann, P., Do-Lenh, S., Bonnard, Q., & Kaplan, F. (2011). Classroom orchestration: The third circle of usability. In CSCL2011 proceedings , ISLS (Vol. 1, pp. 510-517)
- Goligoski, E. (2012) Motivating the learner: Mozillas open badges program. *Access to Knowledge: A Course Journal* 4(1)
- Govaerts, S., Verbert, K., Dahrendorf, D., Ullrich, C., Schmidt, M., Werkle, M., & Law, E. (2011). Towards responsive open learning environments: The ROLE interoperability framework. Proceedings of EC-TEL11, LNCS, (pp. 125-138)
- Hunt, E. F., & Colander, D. C. (2008):*Social science: An introduction to the study of society*, Boston: Peason and Bacon, pp.64-68
- Michel, C., Lavoué, E., Pietrac, L.. (2012) A dashboard to regulate project-based learning. Proceedings at ECTEL12 . LNCS, vol. 7563, pp. 250–263
- Orrill, C. H., Hannan, M. J., Glazer, E.M. (1998) Disciplined inquiry and the study of emerging technology. *Framework*, pages 335-354
- Santos, J.L., Charleer, S., Parra , G., Klerkx, J., Duval, E., Verbert, K (2013). Evaluating the use of open badges in an open learning environment, Proceedings at ECTEL13, LNCS, pages 314-327 (Attached to the deliverable, but this publication was already submitted with D5.2)
- Verbert, K., Duval, E., Klerkx, J., Govaerts, S., Santos, J.L.. (2013): "Learning analytics dashboard applications." *American Behavioral Scientist* 57, no. 10 :1500-1509.
- Wang, F., Hannan, M. (2005) Design-based research and technology-enhanced learning environments. *Educational Technology Research and Development*, 53:5-23.
- Young, J.R. (2012) Badges earned online pose challenge to traditional college diplomas. *Chronicle of Higher Education*