Learner Models in Online Personalized Educational Experiences: an infrastructure and some experiments

Università
della
Svizzera
italiana

Luca Mazzola

Institute of Communication Technologies - Faculty of Communication Sciences

USI - Università della Svizzera italiana

A thesis submitted for the degree of

PhilosophiæDoctor (PhD) in Communication Sciences
major in Technologies

04 2014

PhD defense jury:

1. Supervisor: prof. Marco Colombetti

2. Co-Supervisor: dr. Riccardo Mazza

3. Reviewer: prof. Daniel Burgos

4. Reviewer: prof. Marina Ribaudo

Day of the defense: April 2014

Signature from head of PhD committee:
Prof. Marco Colombetti

Abstract

Technologies are changing the world around us, and education is not immune from its influence: the field of teaching and learning supported by the use of Information and Communication Technologies (ICTs), also known as Technology Enhanced Learning (TEL), has witnessed a huge expansion in recent years. This wide adoption happened thanks to the massive diffusion of broadband connections and to the pervasive needs for education, highly connected to the evolution in sciences and technologies. Therefore, it has pushed up the usage of online education (distance and blended methodologies for educational experiences) to, even in lately years, unexpected rates.

Alongside with the well known potentialities, digital-based educational tools come with a number of downsides, such as possible disengagement on the part of the learner, absence of the social pressures that normally exist in a classroom environment, difficulty or even inability from the learners to self-regulate and, last but not least, depletion of the stimulus to actively participate and cooperate with lectures and peers. These difficulties impact the teaching process and the outcomes of the educational experience (i.e. learning process), being a serious limit and questioning the broader applicability of TEL solutions. To overcome these issues, there is a need of tools to support the learning process.

In the literature, one of the known approach to improve the situation is to rely on a user profile, that collects data during the use of the eLearning platforms or tool. The created profile can be used to adapt the behaviour and the contents proposed to the learner. On top of this model, some researches stressed the positive effects stimulated by the disclosure of the model itself for inspection purposes by the learner. This disclosed model is known as Open Learner Model (OLM). The idea of opening learners' profile and eventually integrate them with external on-line resources is not new and it has the ultimate goal of creating global and long-run

indicators of the learner's profile. Also the representation aspect of the learner model plays a role, moving from the more traditional approach based on the textual and analytic/extensive representation to the graphical indicators that are able to summarise and to present one or more of the model characteristics in a way that is considered more effective and natural for the user consumption.

Relying on the same learner models, and stressing the different aggregation and representation capabilities, it is possible to either support self-reflection of the learner or to foster the tutoring process to allow proper supervision by the tutor/teacher. Both the objectives can be reached through the graphical representation of the relevant information, presented in different ways. Furthermore, with such an open approach for the learner model, the concepts of personalisation and adaptation acquire a central role in the TEL experience, overcoming the previous limits related to the impossibility to observe and explain to the learner the reasons for such an intervention from the tool itself. As a consequence, the introduction of different tools, platforms, widgets and devices in the learning process, together with the adaptation process based on the learner profiles, can create a personal space for a potential fruitful usage of the rich and widespread amount of resources available to the learner.

This work aimed at analysing the way a learner model could be represented in visual presentation to the system users, exploring the effects and performances for learners and teachers. Subsequently, it concentrated in investigating how the adoption of adaptive and social visualisations of OLM could affect the student experience within a TEL context. The motivation was twofold. On one side was to show that the approach of mixing data from heterogeneous and not already related data sources could have a meaningful didactic interpretations, whether on the other one was to measure the perceived impact of the introduction on online experiences of the adaptivity (and of social aspects) in the graphical visualisations produced by such a tool.

In order to achieve these objectives, the present work analysed and addressed them through an approach that merged user data in learning platforms, implementing a learner profile. This was accomplished by means of the creation of a tool, named

GVIS, to elaborate on the collected user actions in platforms enabling remote teaching. A number of test cases were performed and analysed, adopting the developed tool as the provider to extract, to aggregate and to represent the data for the learners' model. The GVIS tool impact was then estimated with self-evaluation questionnaires, with the analysis of log files and with knowledge quiz results. Dimensions such as the perceived usefulness, the impact on motivation and commitment, the cognitive overload generated, and the impact of social data disclosure were taken into account.

The main result found by the application of the developed tool in TEL experiences was to have an impact on the behaviour of online learners when used to provide them with indicators around their activities, especially when enhanced with social capabilities. The effects appear to be amplifies in those cases where the widget usage is as simplified as possible. From the learner side, the results suggested that the learners seem to appreciate the tool and recognise its value. For them the introduction as part of the online learning experience could act as a positive pressure factor, enhanced by the peer comparison functionality. This functionality could also be used to reinforce the student engagement and positive commitment to the educational experience, by transmitting a sense of community and stimulating healthy competition between learners.

From the teacher/tutor side, they seemed to be better supported by the presentation of compact, intuitive and just-in-time information (i.e. actions that have an educational interpretation or impact) about the monitored user or group. This gave them a clearer picture of how the class is currently performing and enabled them to address performance issues by adapting the resources and the teaching (and learning) approach accordingly.

Although a drawback was identified regarding the cognitive overload, the data collected showed that users generally considered this kind of support useful. There is also indications that further analyses can be interesting to explore the effects introduced in the teaching practices by the availability and usage of such a tool.

Declaration

Most of the work contained in this thesis is a slightly modification of papers published in conferences proceedings.

As it is usual in academic community, they are the result of cooperations with different research groups. In this work the two main groups other than my colleague Riccardo Mazza – with whom I worked closer in a day-by-day manner – are the partners in the EU FP7 funded project GRAPPLE directed by prof. Paul De Bra at Tu/E – Technical University of Eindhoven – Netherlands, and the team of prof. Peter Brusilovsky at University of Pittsburgh – Pennsylvania – USA.

The parts included in the present thesis represents mainly original or revised contributions of the author as elaborated for the following published papers.

The publications produced in the context of this study are fully listed here:

- Falakmasir M., Hsiao I., Mazzola L., Grant N., Brusilovsky P. (2012). The Impact of Social Performance Visualization on Students. Proceedings of the 12th IEEE International Conference on Advanced Learning Technologies and Technology-enhanced Learning, pp. 565- 569, Rome, Italy, July 4-6, 2012.
- Mazzola L., Mazza R. (2011). Visualizing Learner Models through data aggregation: a test case. Red-conference, rethinking education in the knowledge society, Monte Verita, Switzerland, March 7-10, pp. 372- 380, ISBN 978-88-6101-010-9.
- Mazzola L., Mazza R. (2010). An infrastructure for creating graphical indicators of the learner profile by mashing up different sources. In Proceedings of the International Conference on Advanced Visual Interfaces (AVI '10), Giuseppe Santucci (Ed.). ACM, New York, NY, USA, 329-332. http://doi.acm.org/10.1145/1842993.1843054

- Mazzola L., Mazza R. (2010). GVIS: A Facility for Adaptively Mashing Up and Representing Open Learner Models. "Sustaining TEL: From Innovation to Learning and Practice" Lecture Notes in Computer Science, Volume 6383/2010, pp. 554-559, Proceedings of EC-TEL 2010, DOI: 10.1007/978-3-642-16020-2_53.
- Mazzola L., Eynard D., Mazza R. (2010). GVIS: a framework for graphical mashups of heterogeneous sources to support data interpretation. 3rd Conference on Human System Interactions, 2010. HSI '10. GVIS: A framework for graphical mashups of heterogeneous sources to support data interpretation. 3rd Conference on Human System Interactions (HSI) 2010, pp.578-584, 13-15 May 2010 DOI: 10.1109/HSI.2010.5514511
  URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5514511
- Mazzola L., Mazza R. (2010). GVIS: An Integrating Infrastructure for Adaptively Mashing up User Data from Different Sources. Information Visualization (IV), 2010, 14th International Conference, pp.68-72, 26-29 July 2010 DOI: 10.1109/IV.2010.19
  URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5571358
- GRAPPLE Deliverable 4.5b : Implementation of interactive visualization of models and students data.
  Public deliverable, available online at http://grapple.project.org
- GRAPPLE Deliverable 4.5c : Evaluation and refinement of interactive visualisation of models and students data.
  Public deliverable, available online at http://grapple.project.org
- Mazzola L., Mazza R. (2009). Supporting Learners in Adaptive Learning Environments through the Enhancement of the Student Model. Human-Computer Interaction. Interacting in Various Application Domains, 13th International Conference, HCI International 2009, San Diego, CA, USA, July 19-24, 2009, Proceedings, Part IV. Lecture Notes in Computer Science 5613 Springer 2009, ISBN 978-3-642-02582-2: pp. 166-175.
- Mazzola L., Mazza R. (2009). Toward Adaptive Presentations of Student Models in eLearning Environments.. 14th International Conference on Artificial Intelligence in Education, AIED 2009, July 6-10, 2009, Brighton, UK. Frontiers

in Artificial Intelligence and Applications 200 IOS Press 2009, pp. 761-762, ISBN 978-1-60750-028-5.

- Mazzola L. (2009). Towards Adaptive representations of Open Learner Models. Doctoral Consortium of UMAP2009.
  URL: http://umap09.fbk.eu/sites/umap09.fbk.eu/files/paper_168.pdf

- Mazza, R., Bettoni, M., Fare, M., Mazzola, L. (2012). MOCLog: Monitoring Online Courses with log data. In: Retalis, S., and Dougiamas, M. (Eds.). 1st Moodle Research Conference Proceedings, Heraklion, Greece, Sept. 14-15, 2012. ISBN: 978-960-98516-2-6

Acknowledgements

The work in the GRAPPLE project was partially supported by EU FP7 funds under the grant #215434: the author would like to thank all the project partners for the fruitful and interesting cooperation.

The work done with Pittsburgh University was partially supported by the Intelligent System Program of the School of Arts & Sciences at University of Pittsburgh. This work was also partially supported by USA National Science Foundation grant DUE-0840597.

I would like to thanks prof. Lorenzo Cantoni for letting us use the Moodle course example as one of the testbed. A special mention goes to Luca Marchesi for having provided me with so valuable ideas and for the help in the implementation of the infrastructure.

Dedication For accomplish this job, I have to thanks a lot of people:

- First of all, **my parents Egidio and Maria Rosa** that supported me in every choice I did, even thou sometimes they did not agree with me. A big thanks goes also to **my sister Lara** that was every time I needed available and happy to pay attention to my problems and ready to give a helpful suggestion.
- From the academic and personal point of view, I want really to thanks **my supervisor**, prof. Marco Colombetti and **my co-supervisor** Dr. Riccardo Mazza, for providing me with so valuable indications and helping me to look over after the momentaneous failure, once received a bad review or in front of an obstacle.
- Without the love and support of my wife-to-be **Liselotti**, I would have for sure not consider to finish my PhD, for these reason (and uncountable others) I would to express my love and gratitude to her.

- For our friendly discussions and for all the support showed, I want to deeply thanks the **colleagues at ITC - Institute of Communication Technologies**, dr. Nicoletta Fornara and dr. Davide Eynard, with whom I also have had the pleasure to write some scientific work and who taught me to apply a scientific methodology.
- To my friend **Federico** for all the experiences that our long-run friendship gave us, for being always enthusiastic at innovative proposals and ready to provide alternative and intriguing ideas, for being the one that opened my mind about opportunity outside Italy and, last but not least, for being the one who reviewed the full thesis and corrected my English mistakes.
- For the other PhD students in the Red-Ink doctoral school at USI (**Emanuele, Silvia, Andreas and Chrysa**), for the useful - and sometimes strong - discussions we had: they are one of the seeds for the current work. A big thanks also for the reciprocal support we expressed in front of all the - also personal - issues that a PhD force you to face.
- To the **staff at Universitá della Svizzera italiana**, prof. Lorenzo Cantoni, dr. Luca Botturi and dr. Isabella Rega and all the staff from **University of Sankt Gallen** and **EPFL - Ecole Politecnique Federale de Lausanne** involved in the Red-Ink school, for the support and indications provided in the development of the PhD research.
- A big thanks to all **my friends** that supported me and was open to share their thought with me, especially when I needed more their support.
- For the beautiful experience of teaching and the very exciting and useful experience in research, my gratitude goes to **my colleagues at Politecnico di Milano**, prof. Francesco Pinciroli, dr. Stefano Bonacina, dr. Sara Marceglia and dr. Simona Ferrante.
- To all **my colleagues at eLab**, **NewMine**, and **WebAtelier** at USI, that shared the physical place in the offices and some projects in this long-run experience that was my PhD; for all the objectives reached together - even if sometimes with difficult and different point of view - thanks to all of you.

- To all the researcher involved in the **GRAPPLE project**, for the long but very helpful and fruitful discussion we had and all the brilliant ideas provided, some of which is my fault not having realized.

This list is not exhaustive; in fact this work was possible thanks to a lot of people that I met along the last four years, and that I want to publicly thanks for all the comments they provided to me. For sure, I let out someone who I deserve gratitude and for this I want to ask sorry in advance.

Contents

List of Figures

List of Tables

Chapter 1

Introduction & Aims of the work

This chapter offers an introduction to the field in which the present thesis is concerned with. After a general overview of the impact that the adoption of Information and Communication Technologies (ICT) has on the educational field, the context of the work is explored. As well as presenting the main ideas and the aims of the thesis, the research topics that represented the basis for the development of the experiments are also described.

1.1 Introduction

Technologies are changing the world around us, and education is not immune from its influence. The diffusion of digital instruments and tools in classrooms, as well as in every kind of teaching and learning experience for that matter, is quite evident to practitioners. Most teachers and students use PCs, tablets and smartphones on a daily basis to search for news or educational contents and more in general use the capabilities of ICT to manage information and produce documents. Because of the above it is no longer possible to consider an educational experience as such without taking into account the digital context in which it is taking place. This is certainly true even when the teacher is promoting a very traditional, paper-and pencil driven approach, as it has been observed that learners still rely and use everyday digital tools to support and enhance their learning.

Another crucial aspect that has to be considered is the wide adoption of distance and blended methodologies for educational experiences, which are enabled and supported by the diffusion of digital tools and media. Unfortunately, besides these positive aspects, the broader diffusion of digital-based educational tools comes with a number of downsides, such as possible disengagement on the part of the learner, absence of the social pressures that normally exist in a classroom environment, difficulty or even inability from the learners to self-regulate and last but not least depletion of the stimulus to actively participate and cooperate with lecturers and peers, i.e. to engage in 1-to-1's and teamwork activities.

For all of the above, the requirement for such tools to be able to effectively support the learning process is a well known, outstanding issue in the field of Technology Enhanced Learning (TEL). The latter is aggravated by the limitations existing on communication channels that are related to the mediation by technologies and the contextual removal of physical co-presence (Kehrwald 2012). Two different ways of improving this situation can be explored which are based on the pedagogical model - to either proactively support self-reflection in a more self-regulated learning context, or to foster the tutoring processes in a way that allows for the learners' activities to be properly supervised. Both could be effectively enabled through the creation of graphical representation of relevant information, i.e. data collected around the teaching and learning process - number of interactions, quiz results, access to resources etc. The aggregated information can then be presented to the users in different fashions based on their profile and role.

The user profiles are normally created using the data collected during the use of the eLearning platform - the methodologies to gather information regarding the users and their interaction with the software have in fact reached a significant level of maturity and reliability in the field of computer science (Fischer 2001). These procedures were originally designed to support developers in discovering bugs or to analyse the user interaction with the system in order to improve the overall quality of the application. Procedures range from collecting explicit user information to gathering user behaviors which is then stored in log files. The latter has become less common due the diffusion of the personal computers, which make it harder to collect and distribute user data from personal devices.

Nowadays though, thanks the massive diffusion of broadband Internet connections and thus the ability to develop programs as web application, collecting user data in the form of interaction logs is back in fashion and is in fact considered as a very valid approach for profiling users. Because of the above, taking advantage of those Web Server functionalities that collect users' navigation footprints (Hoppe, Ogata, and Soller 2007) or including explicit mechanisms that collect user interactions – even at a higher level of aggregation – is finding new interest, particularly in field of Education Data Mining whose purpose is to parse data-intensive fields and identify the patterns and rules which are implicitly contained in the data itself. The ultimate aim of such an operation is to generate forecasts or extrapolate and identify recurring patterns (Ferguson 2012) and (Bienkowski, Feng and Means 2012).

The above is particularly important in relation to the fact that most part of Learning Management Systems (LMS) are web-based (Mazza, Botturi, and Tardini 2006). In fact, in this context tracking user behavior in the form of data log from Web applications is crucial to provide support to both learners and teachers, and can with reason be considered one of the most important data sources to feed monitoring tools with (Mazzola and Mazza 2009a).

A monitoring tool is the system component designated to provide relevant information on a particular activity. An example is the set of led lamps on a computer case - it is designed to give the user an indication of the state of specific hardware components (e.g. hard disk activity, network activity, etc). Logs, on the other end, are normally used to track users' activities. Log-based applications work through a process of specific events aggregation which allows to recognise a complex human activity - or task - within one or more predefined ordered sequences of elementary activities. Each single entry in a web log refers to an individual event. It doesn't contain a great deal of information in itself, but when linked to others logs it makes for a very powerful way to identify the user context. Example of this can be found in (Courtin and Talbot 2006) and (Mazza, Bettoni, Fare, and Mazzola 2012).

Through the aggregation of information into a single log, based on a model developed by experts, a user model is created. These models are normally stored internally within the application without the possibility for the user to scrutinise their personal information. The existing literature suggest that opening profiles to user inspection, with the so called "Open Learner Model" approach (OLM), means the possibility of being able to provide information to the learner about his personal status as derived by the single application, as well as how his/her actions and interactions are interpreted by the system itself (Dimitrova 2008), (Tanimoto, 2005), and (Bull and Nghiem, 2002). Other critical issues could also be address by opening personal profiles. For example, it would be easier to identify problems or assess a lacks of precision in the profile itself, simply by allowing user inspection (Bull and Kay 2007) and (Shahror and Bull 2008).

The idea of opening learners' profile and eventually integrate them with external on-line resources is not new and has the ultimate goal of creating global and long-run indicators on the lerners' profiles. There are different possible approaches to make the profile scrutable by learners, in fact in the literature are well known the approaches based on textual presentation but many researchers moved towards graphical indicators, considering it as more effective and natural for the learner and the actual devices where the courses are offered. The generated objects, called smart widgets by some researchers (Glahn, Specht, and Koper 2008) – are simple graphical objects that aims at exposing valuable information to the user. This will take into account internal and external user data in order to have a more complete and comprehensive view of the user's behaviors in the eLearning tools used.

On top of the above it has to be considered that another element plays a major role in this scenario: the paradigm of Web 2.0. This has in fact changed not only the way we explore and search in the Internet sea, but also how users expect to interact with online resources. Posting comments about news in a blog, refining an article in a collaborative wiki, or aggregating information from heterogeneous sources are features that also affect the eLearning field.

The sum of all these types of activities widely changed the concept of the Computer Mediated Education, forming the basic bricks for, and enabling a wider definition of Technology Enhanced Learning (TEL) as the evolving virtual place for formal, semi-formal, and informal learning. The shift in the model enforced by these interactive tools impacted heavily on the organizational aspect as well.

Despite some initial attempts to accurately reproduce the face-to-face experience in an online environment, a new approach has recently emerged. This relies more on the peculiar characteristics of the electronic medium than the traditional non-digital tools and pushes the capabilities that the technologies can offer. This shift allowed to reach the objective of providing a better educational experience (in the sense of better informed, more available and flexible) while giving the possibility to target and tailor the interactions and educational experience to the learner needs.

As a consequence, the introduction of different tools, platforms, widgets and devices in the learning process creates a personal space for a potential fruitful usage of the rich and widespread amount of resources available to the learner (Klebl, Kramer, Zobel, Hupfer, and Lukaschik 2010), (Ebner, Holzinger, Scerbakov, and Tsang 2011), and (Conole 2012). In fact, the availability of many contents and activities not specifically designed to be part of a single pre-structured and inflexible didactic flow enriches the learning experience, offers teachers the possibility to prepare additional and alternative paths and also opens up to the introduction of some Informal Learning activities, thus demonstrating its potential full impact.

Furthermore, with such an open approach, the concepts of personalisation and adaptation acquire a central role. On the personalisation side, the possibility to choose amongst different options such as the type of media used (text, audio or video) or the approach adopted in the subject presentation (inductive or deductive) allows learners to enjoy an experience which is better-suited to their stated preferences. On the adaptation side, the students are – either fully or semi-automatically – provided with the content that is more appropriate to their profile. The integration between dynamically added sources, heterogeneous tools, devices to support different situations, combined with the two processes of personalisation and adaptation, eventually can create an ideal working space for adaptive learning systems.

Researches in this context have demonstrated that it is useful to consider the richness of the experiences adopting a holistic approach (McCalla 2004), as it it usually the case with complex system e.g. a natural ecosystem in ecological analysis. On the adaptive features' side one of the most important components is the student model, which is also in charge of keeping track of the learner's knowledge and skills acquired during the learning process. As already discovered (Bull and Kay 2008), in order to increase the level of engagement of learners, to support their consciousness of the current status and to encourage reflection as learning (Bull 1997), the student model could be opened to the inspection of learners and instructors. Student models are usually made available as visual representations, because it simplifies the data interpretation by human beings (Scaife and Rogers 1996) and (Ferreira de Oliveira and Levkowitz 2003). Since the user information is not only stored into a specific student model system but also in this new complex ecosystem, it is often distributed in a number of platforms used for different purposes (e.g. LMS data, intranet usage data, resources access and so on) and therefore data must be aggregated from different tools and provided consistently to the interested users, preferably in visual formats (Dror, Nadine, and Mike 2008).

In the literature, other approaches to the problem of creating Open Student Models in TEL are "educational mashup" (Esposito, Licchelli, and Semeraro 2004) and "ubiquitous and decentralized user model" (Van Der Sluijs and Houben 2006) and (Heckmann, Schwartz, Brandherm, and Krner 2005).

Some issues however arise from the adoption of these approaches. In fact, from the user point of view we have to consider that a high quantity of mashed-up data might cause an overload problem (Chen 2009) which may become problematic if it ends up distracting from the learning activity and confuses the learner about the represented data (Ahmad, Basir, and Hassanein 2004) and (Costabile, De Marsico, Lanzilotti, Plantamura, and Roselli 2005).

A possible mitigation of this issue could be implemented using a minimal impact policy i.e. presenting a compact, general and summarising indicator in the eLearning tool interface that works as the access point to a specialised dashboard. The dashboard collects a set of visualizations provided to the users for an autonomous exploration on and in-depth analysis. This approach also allows to stress the message that learners can use this additional tool for better understanding the personal situation, but that this information is only for self-assistance and does not represent the subject or the main objective of the educational experience.

On the other side, in order to mitigate the overload issue, the visual representations can be made adaptive to the role, to the context and to the activities performed by the learner. With this kind of approach, the adaptation helps creating easier and more understandable indicators. For example the adaptive dashboards, widely spread in the field of Business Intelligence, (Schutz 2009) are used to represent the most useful and relevant subset of the all information available for an ongoing task, without restricting the users for enjoying a more in-depth view on specific data, based on their personal interest or other criteria.

The next section will describe the context of the work, including the aims of the research and the research questions. Methodology adopted will also be explained.

1.2 Context of the work

The present section explores the scope of the work and is divided in two parts. The first presents the basics ideas and underlying concepts while the second focuses on the research questions as well as the methodologies used to investigate them. The starting point of this research are some of the requirements identified in an EU FP7 funded project called GRAPPLE, which will be explained more in detail later on[1].

In this context, the collection of interaction data from learners about activities, preferences and characteristics is a central as well as mandatory task for providing students with a personalised experience based on adapted contents. The possibility to disclose these data for supporting users is well known in the literature (Dimitrova, Self, and Brna 2001), (Bull and Nghiem 2002), and (Bull and Kay 2007) and seems to have effects on self-reflective processes (Tanimoto 2005) and (Gama 2004).

1.2.1 Ideas

In the TEL context, the creation of a user profile is fundamental for fulfilling some didactic tasks, such as measuring the degree of participation to a course, the performance around quizzes and assignments or the degree of commitment to a discussion forum / blog. The procedure involved with the profile generation is based on the collection of information usually stored as a number of log files generated by software applications. The tasks dedicated to user profiling are normally already managed by LMSes, which also provides a way for exploring the collected information. There is however no way, at present, to target the data usage and presentation or personalise it based on "semantic" data interpretation. The purpose of this work is therefore to shape and adapt the visual presentation of the data based on one or more of the different characteristics defining the learning experience (i.e: the user role, the context, and the device used).

With such an approach, it also becomes possible to think about integration of data coming from different systems. This possibility assumes a higher relevance if we consider the Life Long Learning (LLL) context, which is the process of personal continuous enhancement and empowerment that takes place on an on-going basis from our daily interactions with people and environments. One of its peculiar characteristics is the use of different systems in different institutions across the whole learning process and the generation of 'isles' of data that need to be connected in order to create a unique and possibly valuable learner's profile. To overcome

[1]GRAPPLE project website - available online at: http://grapple-project.org

the limitations that these 'isles' imply, studies are being devoted to the enhancement of the procedures related to users' profile creation, both in term of data source to be used and meaningful integration of heterogeneous sources (Abel, Herder, Houben, Henze, and Krause 2010), (Abel, Henze, Herder, and Krause 2010), and (Leonardi, Abel, Heckmann, Herder, Hidders, and Houben 2010).

The idea of opening learners' profiles, maybe also for integrating them with external on-line resources, has the ultimate goal of creating graphical indicators for the profiles themselves. The generated smart widgets (Glahn 2009) and (Glahn, Specht, and Koper 2008) – simple graphical objects that aim at exposing valuable information to the user – will take into account the highest possible amount of user data, in order to have a more complete and comprehensive view of user behaviors in Learning Management Systems. This data can come from the LMS itself (internal) but can also be collected from other sources (external).

At the same time, this data can be used to create a dashboard for monitoring the status of a course: in this way teachers and tutors could be supported in achieving a better understanding of the current status of the class and, possibly, in adapting the resources and the pedagogical approach accordingly with this information.

Additionally, the Instructional Designers (IDs) (i.e. the pedagogical experts, who are in charge of specifying the didactic interpretation of the stored data in a specific educational environment), can be supported by the awareness of the effective usage of resources and the understanding of which kind of activities are widely adopted, in order to better target their teaching.

Platform administrators can be also supported by a different use of the same information. In fact, a compact representation of some users behaviors in the system, e.g. the login frequency, the preferred time of usage and the kind of resources/activities mainly used on the platform, is valuable information to correctly set up the hardware and software requirements and to monitor the correct platform behavior at run-time. Furthermore, manually extracting this type of data in a just-in-time fashion is normally a difficult, time-consuming, and error-prone task. An automated approach would therefore reduce risk while increasing performance and accuracy.

1.2.2 Aims

This work aims at investigating how the adoption of a tool for offering adaptive visualisations of Open Learner Model to the student could affect the user experience in an educational system. Both the learner's and the teacher's sides were investigated in order to achieve a global vision on what sort of impact would be generated by its introduction. Part of the job was also devoted to exploring how the introduction of adaptivity – in the amount of data presented as well as the information encoding paradigm and the interface – affects its perception.

The analysis was mainly based on two different levels: the possible effects reported by users after the analysis of the mock–ups and the interaction with the system itself (self–evaluations), but also on run-time evaluations when feasible, as would be the case for the learner's side. Nevertheless, when this was not possible, we relied on feedbacks, provided by final users, and their opinion about the possible impact they expect.

After the requirements were identified, the research in the GRAPPLE project continued with the development of a general and configurable tool, called GVIS (**G**RAPPLE **VIS**ualization tool), to extract, aggregate and provide information to users of online educational experiences.

The development was carried out – for the sake of portability and applicability – using a very generic approach, i.e. decoupling the technical infrastructure from the data semantics, stored into configuration files. This approach allows the software to be completely independent from a specific source or a system using the elaborated information (and in this way to work smoothly with different Learning Management Systems or environments), but also support it to be easily portable between different platforms.

Furthermore, the IDs can rely on these configurations to speed up the development of their own aggregation models without the need of programming skills. The aggregation is the model that drives the creation of indicators based on the data interpretation.

While specifying a new aggregation template is a task that requires some capabilities and attention, it also allows for full customisation. This can empower the ID in defining new cases for data usage through indicators that better fit in with the designed didactic experiences. The collaboration with experts in the field of pedagogy was fundamental for developing such an approach. In fact, it allowed to create new general configurations to be used with the tool based on different pedagogical models. It was also crucial in the interpretation of the quantitative information that was collected in order to validate the application of the GVIS tool.

The work had two main contributions. The first was aimed at showing that the approach of mixing data from heterogeneous and not related data sources can have a meaningful didactic interpretation. This can be achieved through an explicit declaration of the data semantic for the educational experience.

The second contribution was more related to the perceived impact on online experiences of the introduction of the adaptivity in the graphical visualisations produced by such a tool, especially applied to a LLL approach.

An optional objective was the explicit identification of good practices and criteria, if such things even exist, to develop templates for the implemention of services for learner models, which could also be empowered by adaptive functionalities. Unfortunately, this was not fully achieved due to the difficulties in finding teachers with different didactic approaches keen to take part in the experimentation.

For the first contribution, a number of test cases were created to demonstrate the portability of the tool among different systems – mainly educational tools – and the possibility to offer valuable information from heterogeneous and distributed sources.

The second contribution was analysed through a measurement of the impact of OLM adaptive representations on the learner's online educational experiences and through the perceived impact on teacher. The plan was to also measure it through the level of self-confidence (Bandura 1997) and (Tschannen-Moran and Wolfolk Hoy 2001) provided by the adoption of such a tool but, as already mentioned, the difficulty in finding enough teachers willing to participate in the experiments made this non-achievable.

The third potential contribution was directed towards the identification and classification of the criteria for developing an adaptive learner model. This could have been a more theoretical contribution to the field. Creating such a taxonomy could speed up the adoption of this kind of services, by showing the possibility to seamlessly integrate some adaptive behaviors in already existing externalised models. This was also expected to justify the previous objective and to propose this open problem to the community for further investigation and research.

1.2.3 Research Questions and Methodology

After demonstrating the generality of the tool and the applicability of the semantic data description approach, the main question driving this research was how an adaptive visual representation of the Open Learner Model could improve the user involvement in eLearning experiences without overloading the cognitive aspect.

In order to find an answer, it was first of all necessary to understand how a learner model could be represented in term of visual presentation to the system users (A). Subsequently, its effects and performances for learners (B.1) and teachers (B.2) could be analysed. Throughout the research process to investigate the above, an additional issue was raised which had to be solved: could general criteria for building adaptive model in the visualisation of OLM be defined (C)? Most of the Open Source (OSS) eLearning systems that are currently available do not internally provide a comprehensive and life-long learner's model based on the definition given above.

Thus, as a prerequisite for the experiments, aggregation templates had to be defined to support the creation of a user model in the learning environment, starting from heterogeneous sources among which a primary role was once again played by the logs collected from the learning management platforms in adoption.

This could be achieved – in each tool that has to be connected – through the semantic interpretation, inside GVIS, of logs representing users' events recorded in the educational platform and integrated with data coming from other tools used in the educational experiences. This means the ability to give a meaningful didactic interpretation to the actions performed or the status achieved by users.

Point A (about possibilities to visually encode the learner model for presentations purposes) is based on the collection of educational templates that are a good fit for the online experiences used for runtime testing. The educational templates are sets of instructions codified in a formal language that describe the data transformation and the information aggregation that make sense for the didactic approach adopted by the ID. This set was created by reusing the templates already developed in GRAPPLE[1] or through the identification of fresh ones for each individual case. The set of templates chosen for GRAPPLE by researchers – based on stakeholder interviews – was instantiated in the GVIS tool and adapted to run the experiments. In this phase a fundamental contribution from pedagogues/teachers of the university offering the course of interest has assured the adherence of the aggregation templates to the didactic approach used in the online course.

A positive side effect was the extension of the set of pre-defined configurations that was developed inside GRAPPLE. These configurations will be released and a part of them are attached to the current thesis, as examples, in the appendix.

For the objective B (effects and performances analysis), a quantitative analysis was chosen as the most suitable option. Point B.1 (for learners) was analysed referring to Kirkpatrick model (Watkins, Leigh, Foshay, and Kaufman 1998) and exploring the first two levels defined in the stacked model presented which refer, respectively, to the reactions to the introduction of

[1] GRAPPLE stands for "Generic Responsive Adaptive Personalized Learning Environment". The GRAPPLE project aims at delivering to the learners a technology-enhanced learning (TEL) environment that guides them through a life-long learning experience, automatically adapting to personal preferences, prior knowledge, skills and competences, learning goals and the personal or social context in which the learning takes place. The same TEL environment can be used/accessed at home, school, work or on the move (using mobile/hand-held devices).

the tool and to the effects on knowledge produced by its usage. The methodology applied was based on online questionnaires (an effort was made to try and keep them compatible with the ones used in the initial evaluation of the GRAPPLE project, in order to allow for comparison) and analysis of the performances of the courses in term of pre/post test (or, when not available, of self-evaluation one) and grades received by student, if available for the inspection.

For objective B.2 (effects and performances for teachers), the original plan was to apply a sub-part of the Teacher Self-Efficacy (Bandura 1997) and (Tschannen-Moran and Wolfolk Hoy 2001) questionnaire, in order to allow the emergence of the possible empowerment offered by the usage of the GVIS tool in the teaching practice of eLearning experiences. Even though the small amount and insufficient heterogeneity of the teacher participating in these experiments has not provided us with statistically relevant analysis, we offer some initial considerations based on the self-evaluation and perception of impact from teachers and tutors involved, collected through online questionnaires.

Finally, point C – related to the idea of shared processes able to support the design of templates for creating adaptive visualisation of OLM – was not analysed separately but rather viewed as a milestone for running the experiments. No general criteria emerged from the work that was carried out. To tackle this point we had to rely solely on the professional ability and discretion of the Instructional Designers as the adoption of an explicit external design was not a doable option.

This work continues as follows. Chapter two explores the state of the art in the field, analysing the areas of IV, TEL, OLM, and EDM (with some additional consideration about the possible impact of externalisation). Here the position of this work is also presented in relation to other researches. In chapter three, the context in which the framework for the creation of user indicators (called GVIS) initially took shape is presented. Chapter four deals with the technical implementation of the software tool. Chapter five is around experiments that were performed using the tool both inside different LMS and Intelligent Tutoring Systems (ITS) as well as stressing data source coming from different context of activity. In this chapter we also offer some initial evaluations of the tool, in terms of adherence of the designed functionalities to the perceived needs on the one side and in terms of the effects generated when applied in study cases on the other one. Chapter six is composed of three sections: the first one discusses objectives that were achieved as well as problematic areas or aspects, the second one draws the overall conclusions, and possible next steps and references are presented in the final section.

Chapter 2

Background / State of the Art

The background of the current work can be identified as a set of basics thematic areas which provide, as an overall, the bricks for meaningfully contextualise the experiments done using the tool developed based on the user requirements. In the section dedicated to Information Visualization, the usage of techniques and approaches for representing a rich and complex set of information in a compact and effective ways is explored. In the Technology Enhanced Learning section the broader context is presented and further expanded – primarily around the main subject – in the paragraph about Open Learner Models. The reasoning as well as the association of the data to the semantics is discussed in the area of Educational Data Mining, where concepts around Learning Analytics (LA) are also described. A description of the possible effects of the externalisation of the learner model is then presented. Finally, in an additional section of the chapter, a short introduction to the objectives of the Grapple project is included as well as the specific context in which the GVIS tool was originally conceived, planned and developed.

2.1 Information Visualization (IV)

Information Visualization is the field of Computer Science that examines techniques for representing a vast amount of abstract data in a visual format, so that the data can be comprehended and interpreted by human beings. It is also defined (Card, Mackinlay, and Schneiderman 1999) as the art of putting together small data fragments, that taken alone have no real value or usage, to create a graphical representation that can enhance the reader visual system for the knowledge processing and acquisition. Binding on this premise, the main goals of visualisations as reported in the literature can be divided into three categories: exploration (researching relationships, trends, and interesting phenomena); confirmation (validating or refuting hypotheses); and presentation (conveying information to others) (Spence 2007).

In the educational context, several researchers use visualisation techniques to provide tools to support more effective learning and instructions (Duval 2011). However, while these visualisation tools could be valuable for both instructors and learners, the majority of research in that field targets instructors and educational institutions (Dawson 2010), (Dawson, Bakharia, and Heathcote 2010), (Graf, Ives, Rahman, and Ferri 2011), (Vatrapu, Teplovs, Fujita, and Bull 2011), and (Zhang and Almeroth 2010).

Dawson (2010) proposed a model for capturing and analysing the changes in students behaviors and their learning network composition for the purposes of proving educators with visual information to support their intervention, especially to the ones identified as being “at risk”. The availability of such a tool – through the usage of the SNA[1] applied to engagement data – allows instructors to make more reasoned and informed choices about their didactic plan and its evolution.

In the work of Dawson, Bakharia, and Heathcote (2010) a model that tries to recover the multiple learning hints lost in the online educational experiences in respect of the face–to–face ones was instantiated in a tool called SNAPP (acronym for Social Network Adapting Pedagogical Practices). The authors demonstrated that, by stressing the new computational and storage capacity of the recent IT infrastructures available, the implementation of learner model became affordable in real–time, this being one of the condition that was previously preventing the adoption of this kind of approach. The visualisation of this real-time evaluative data – such as indicators of social network structure and centrality, social interactions and communication flows representations, connection degrees of learners (nodes), and hub structure inside the eLearning community – was then able to support the activity of educators both in intervening for supporting the learners and in better planning didactic activities.

Graf, Ives, Rahman, and Ferri (2011) concentrated their attention on the design of a DSS[2] that would overcome the availability of reports based on the very general and limited information normally provided by the learning management systems adopted by educational institutions.

[1]SNA is an acronym for Social Network Analysis and is an approach consisting in extracting information about the relationship that exists in one environment relying on the links that connect its elements.

[2]DSS stands for Decision Support System which is a typology of software devoted to the aid of the human decision process, through the offer of additional information, its aggregation at a higher level or the extrapolation of cubes inside the data.

They relied on the extracted information about the learning process to support teachers and course designers in identifying difficulties or inappropriate learning materials. The main contribution they expected to provide was related to the design of improved educational resources and supporting activities based on the students behaviors during their usage and delivered through visual hints.

Vatrapu, Teplovs, Fujita, and Bull (2011), on the other end, focussed on the teacher's dynamic diagnostic process, as they consider this skill of primarily importance in the teaching profession. They developed a triadic model of the "teaching analytics" – called *TMTA* – which is based on the collaboration of a Teaching Expert (*TE*), a Visual Analytic Expert (*VAE*) and a Design–Based Research Expert (*DBRE*). The model can be used for planning and evaluating an instance of the classroom activity. With regards to the application of this model they proposed a strong coordinated action of these three experts to analyse, interpret and act upon the real-time visual information (info–graphics) extracted from the learners' interaction with the educational platform. Their work does not offer actual visualisations, but rather proposes a model that has to be instantiated –together with the visualisations – during the planning of the classroom.

Zhang and Almeroth (2010) analysed the cases in which Information Visualisation is used to create indicators of the learning activities in conjunction with a well known LMS that provides only a very limited and fixed set of information about the students' activities, not fully exploiting the richness of the collected log. The tool is tailored to instructors and educational researchers to help them evaluate the contribution of the LCMS [1] to the learning, through the assessment of the learners' behaviors and progresses. The present work tries to extend this approach, making the data extraction and aggregation process independent from a single specific LMS.

Only a minor fraction of projects focus on providing visualisation to students (Arnold and Pistilli 2012), (Arroyo, Ferguson, Johns, Dragon, Mehranian, Fisher, Barto, Mahadevan, and Woolf 2007), and (Long and Aleven 2011).

Arnold and Pistilli (2012) concentrated their attention on the development of an early intervention solution to provide real-time feedback to students. The tool – called *Course Signals* – relies on multiple typologies of data about students e.g. grades, demographic characteristics, past academic history and current effort (as measured by the interaction with the educational platform). It creates an indicator based on color, adopting the metaphor of the traffic light

[1]LCMS stands for Learning Content Management System which is usually adopted as an alternative definition for LMS, based on the fact that historically its main usage was – completely or extensively – just to deliver educational contents, without the addition of any specific educational activity.

signal, and delivers it to each student by means of the institutional email to indicate them how they are doing in respect of the objectives settled by their tutors. Additionally, faculty members receive a personalised report comprising the status of their students: in this way, the adoption of the tool has also the effect of sharing the perception of the situation amongst the different stakeholders in the educational experience.

Arroyo, Ferguson, Johns, Dragon, Mehranian, Fisher, Barto, Mahadevan, and Woolf (2007) reported the effect of disclosure, in the form of graphical indicators, of the interventions put in place by an automated tutor to support the learning experience, such as the progress charts that disclose the evolution of their responses accuracy to the learner. They noticed that students tend to be disengaged after using a tutorised system for a certain amount of time, but directly providing them with self-monitoring functionalities could induce a re-engagement, based on the self-reflection and self-monitoring processes that they will trigger. They stressed this evidence by implementing an Open Learner Model enhanced with explicit suggestions and encouragements, based on the specific real–time situation of each learner.

Finally, in the work of Long and Aleven (2011) they explored the perceived impact and the behaviors induced by the usage of information about the learner profile by the student itself through surveys and interviews with the learners. They offered simple visualisations, based on bar-charts, to represent the level of skill mastered by the learner. The teachers' point of view, obtained through interviews in the analysis phase, is also taken into account and used as a reference point. The authors discovered that the possibility to witness the evolution of this very simple OLM encourages the learning process. Another interesting finding was related to the discrepancy between the model as calculated and stored inside the system and as self-perceived by the learner, which ingenerates in the learners a sentiment of mistrust in the system. What they discovered is that this is reported to happen quite frequently, demanding an intervention to solve this situation.

This second aspect justifies all the researches on interactive open learner models, in which the learner can interact and modify the model itself, sometimes simply by indicating a perceived variation of his/her needs and some other times by proving their competency with question or supporting evidence.

Finally, they observed that the possibility to inspect these models is not automatically accompanied by a self-reflective process, but needs to be specifically supported and induced. This is made possible, for example, by providing a compact visual cue about the current status of the student model and allowing further explorations of the model, through interactions and the reflection on the specifically provided at a higher level.

2.2 Technology Enhanced Learning (TEL)

The field of Technology Enhanced Learning is related to the usage of digital technologies in the practice of education. Its main focus is the alignment between the technologies applied and the different aspects of the learning experience – resources, actions, and objectives – in order to provide socio-technical innovations in education, independently from time, place and pace constraints.

Unfortunately some negative side effects are well known, like a higher rate of dropouts (Levy 2007) a feeling of loneliness, isolation and low motivation to learn (Rovai 2002).

Levy (2007) – after a more formal definition of what can be considered a dropout in eLearning field, which was previously not well defined – explored the possible reasons connected with its increase in online experiences compared to the on-campus presential ones. The main finding was that amongst the key–factors considered –namely the *academic locus of control* and the *student's satisfaction*– only the second one showed to be a reliable indicator, whereas the locus of control seemed to play no role in the student's dropout rate. As expected, the author found a negative correlation between the students' satisfaction and their dropout rate.

Rovai (2002) on the other end explored the impact of learning communities in educational experiences, comparing the cases of presential ones to the eLearning ones and positively correlating the sense of being part of a community with a higher level of fulfilment and satisfaction. This work reached the conclusion that fostering the creation of learning communities in online courses can facilitate the dialog and decrease the psychological distances amongst the participants. In the present work this point supports the idea of extending the presented information to social aspects.

Other authors have also reported that these issues could be reduced by increasing the level of engagement among students, such as in (Laurillard, Oliver, Wasson, and Ulrich 2009). They reported the capacity of the new digital media to connect innovation and practices, generating a natural sense of engagement and curiosity towards the messages encoded on that medium. In fact, the authors stressed the fact that the adoption of the new digital media can be a way to improve the students' capabilities to express themselves, thus allowing them to enhance the expressiveness and creativity in the educational process while scaffolding their intellectual development.

It was also found that an holistic approach demonstrates to be useful in understanding and tackling these issues (McCalla 2004). The author argued that it is fundamental to contextualise the eLearning experience, going further into the *semantic web*[1] approach and introducing a *pragmatic* dimension, where the user context, intentions and objectives can be captured and used to induce a reaction in the proposed experience. The sum of these "pragmatic" layers i.e. the always existing "semantic" level, the user profile and the educational resources create the ecological approach for the design of TEL empowered systems. This holistic approach, which uses all of the defined levels as the knowledge base for the mining process, allows for a better contextualisation of the patterns that emerge inside the learners' educational experiences and provides them with a more engaging experience.

Another possible approach is to automatically adapt the learning experience to some of the learner's characteristics without relying on a pragmatic layer. Some of the dimensions that can drive the adaptation are personal preferences, learning goals, personal and social context or a student model i.e. a profile of the knowledge and skills acquired during the learning process.

Some experiments to create a learner model were conducted by researchers (May, George, and Prevot 2007) in the context of CMC[2] tools (like web discussion forums), where they created a model for collecting the breadcrumbs[3], or in other words procedures for the manipulation of this data to obtain a profile and an approach to securely store the resulting model. Based on this collected raw data, they proposed some common analysis to produce the model which should be also visualised to guarantee optimal usage.

The learner model is also reported to be able to provide some interesting information about the student mental situation, intended as the processes and cognitive functions specifically stimulated by the current learning activity. In fact, through the abstraction process and the *Semantic Web* approach it becomes possible to also consider the user context and provide information better suited for supporting the learners performances in achieving their full potentiality in online tasks (Heath, Motta, and Dzbor 2005).

[1] for Semantic Web here is intended an approach to the information published on the Web that, through the explicitation of the semantic of the hyper-link and the data presented, support the automatic reasoning over this data and the extraction of new knowledge by programmers.

[2] CMC stands for Computer Mediated Communications

[3] *breadcrumbs*, literally the small pieces of bread created when you cut it, which represent the elementary fragments of information left online when the user interact with a web-server.

As stated in the previous section –centered on Information Visualization–, it is commonly accepted that choosing a graphical presentation could make the interpretation of this complex data much easier (Dror, Nadine, and Mike 2008). In this paper they proposed a methodology called *VAF* – Virtual Apparatus Framework – based on a novel visualisation tool *Solution Trace Graph*. With this approach, they were able to develop an intelligently adapted remediation system in an exploratory learning scenario.

Furthermore, opening this model to the student's inspection is another option to increase their level of engagement, stimulating the perception of the current status (as already shown by other researcher), and to encourage reflection as learning.

Bull (1997) proposed a system, called *See yourself Write*, that presents a merged picture of the information about the assignments submitted and the feedback provided by the tutors and discloses the generated model for reflection on the path completed and the feedback received by learner.

This specific aspect will be presented in details in the next section, that deals with the so-called Open Learner Model approach.

While crucially important, OLM is not the only method proposed for solving the problem of creating and memorising a learner's profile that support disclosure functionalities. Other ways to create, maintain, store, and externalise the model gave origin to different approaches, known as "educational mash-up" (Esposito, Licchelli, and Semeraro 2004) or "ubiquitous and decentralized user model" (Van Der Sluijs and Houben 2006) and (Heckmann, Schwartz, Brandherm, and Kroner 2005).

Esposito, Licchelli, and Semeraro (2004) faced the challenge of creating a student profile relying on procedures and approaches from *IR* (Information Retrieval): their *Profile Extractor* uses *ML* (Machine Learning) techniques to discover the preferences, needs, and interests of the learner. The source of data taken into account to extract the information are the learning performances, the communication preferences and the online behaviors adopted by students.

Instead Van Der Sluijs and Houben (2006) proposed a semantic approach to collect data from web application (based on the Semantic Web model) and to retrieve and connect data generated by the same learner inside different platform, as usually happen in the Web environment. They elaborated the *GUC* (Generic User model Component) as an autonomous and pluggable component for this task.

Another approach – which was lately merged into one with the one above – was ideate by Heckmann, Schwartz, Brandherm, and Kroner (2005) who proposed to rely on an ontology developed by themselves called *GUMO* (Generic User Modeling Ontology) and on an extension of XML, called *USERML* to track, store and exchange information about learner profiles, on top of a server to store and query the "global" profiles created.

2.3 Open Learner Model (OLM)

One of the earliest attempts to provide visualisation tools to identify risky students and devise ways of supporting their learning has been done by Mazza and Dimitrova (Mazza and Dimitrova 2007). CourseVis is a visualisation tool that helps instructors to early identify problems students may have.

There are two main independent directions of research on open learner models. One direction focuses on visualising the model to support students' self-reflection and planning. The other one encourages students to participate in the modeling process, such as engaging students through the negotiation or collaboration on the construction of the model (Mitrovic and Martin 2007).

Representations of the student model vary from displaying high-level summaries of the information included in the model (such as skill meters[1]) to complex concept maps or Bayesian Networks[2].

A range of benefits have been reported from the opening of the student models to the learners which range from the increased learner's awareness on the knowledge development process to the elicitation of the difficulties encountered (Mitrovic and Martin 2007). Furthermore, the disclosure seems to have a positive impact on the students' engagement, motivation, and knowledge reflection (Bull 2004) and (Zapata-Rivera and Greer 2004).

Dimitrova, Self, and Brna (2001) explored interactive open student modeling by engaging students to negotiate with the system during the modeling process.

Chen, Chou, Deng, and Chan (2007) investigated active open student models in order to motivate them to improve their academic performance.

Brusilovsky, Sosnovsky, and Shcherbinina (2004) embedded, inside one of their adaptive link annotation systems known as *QuizGuide*, an open learning model in the engine and demonstrated that this arrangement can remarkably increase the student's motivation to work with non-mandatory educational contents.

[1] a *skill meter* is generally a very compact indicator of the mastery level on concepts or skills achieved by learners

[2] Bayesian Networks is an approach to build an explanation of hidden variables by observing the external status, stressing their conditional probabilities and causal relationships

To support social learning, a common approach is to show learners average values of the group model e.g. the average knowledge status of the group in a given topic. These models fall into the category of group based student models. Both individual and group based open student models were studied and the increase in reflection and helpful interactions among teammates through their adoption was demonstrated.

Bull and Kay (2008) described a framework to apply open user models in adaptive learning environments and provided many in-depth examples. Open group modeling enables students to compare and understand their own state among their peers. Moreover, such group models have been used to support collaboration between learners among the same group, and to foster competition in a group of learners (Vassileva and Sun 2007). The authors investigated the role of social visualisations[1] in online communities. They concluded that this kind of visualisation increases social interaction among students, encourages positive competition, and provides students with the opportunity to build trust in others and in the group. Bull and Britland (2007) used their OLM implementation – called *OLMlets* – to investigate the facilitation problem for group collaboration and competition. The results showed that optionally releasing the models to peers increases the discussion among students and encourages them to start working sooner.

The implementation of an Open Learner Model represents a possible solution to bridge the gap between the functionalities expected by learners and the capabilities offered by the system, in terms of interaction possibilities and presentation of relevant information in eLearning systems. It also allows to offer a fully customisable and adaptive interface to the learner's model (Brusilovsky 2004) with respect to the users' characteristics, preferences, knowledge, and tasks (Mazza and Dimitrova 2007).

Student-related data is collected in the student model, which is a component of adaptive systems that maintains an accurate representation of the user's current state, enabling the system to perform adaptation based on the information stored in the model (Mitrovic and Martin 2007). The adaptation of the contents to the user's knowledge and cognitive characteristics (Bull 2004) is a way to support the current learning needs of the learner. It is also generally accepted that it is a well-suited approach to increase the learner's level engagement in the educational experience (Zapata-Rivera and Greer 2004), thus allowing to offer a truly customised

[1] A *social visualisation* is a representation of the traces left by a user in the interaction with the platform and others that enfatises the social purposes, such as information exchanges, cooperation, dialogs, reciprocal position in a network, common and different skills and achievements, etc. These kinds of visualisations can be used to enhance the awareness of one's social environment or to express cues and patterns which are implicit in the underlying communication.

experience. Opening this internal model to user inspection could be useful for different reasons, and in particular for self-reflection (Dimitrova, Self, and Brna 2001).

In this view, the model is also an useful source of information that can be used to reinforce the user's commitment to the online experience and to foster his/her self-reflection processes (Chen, Chou, Deng, and Chan 2007).

More recently some attention has been devoted to the aspect of social interaction supported by online platforms, and the relative representations provided by the systems have also been modified accordingly (Brusilovsky, Sosnovsky, and Shcherbinina 2004). The possibility to include data from external sources could empower the profiling mechanism in having a model that also caters of social and affective characteristics of the learners (Vassileva and Sun 2007) and (Bull and Britland 2007).

2.4 Educational Data Mining (EDM) and Learning Analytics (LA)

The creation of OLM requires the distillation from a huge amount of raw data of information and knowledge about one or more characteristics of the learner, such as preferences, interaction habits, knowledge, skills,and experiences. This task could be supported by data intensive techniques, such as those developed in the Data Mining (DM) field.

DM techniques comes from fields such as economy or marketing, where they are used to find the most common pattern or co-occurrence in the buying habits of consumers. This approach generates rules in the form of *precondition* => *postcondition* that are not really devoted to investigate the causal relationships generating the rule itself, but which is more interested in the *coverage* (how many buyers amongst the sampled ones follow that specific pattern) and *support* (what percentage of the buyers that have the *precondition* also have the *postcondition*) of the found rule (Agrawal, Imieliński, and Swami 1993).

When specifically applied to the field of education, this approach takes the name of Educational Data Mining (Romero 2011). The above has been defined as a separated field due to the specificity of the kind of rule involved and the particular attention paid to the learning domain, peculiar of its own. In fact, this is especially relevant in order to achieve a better learners' understanding, but also to explore and offer an in depth interpretation of the learning context [1].

[1] AA.VV, "EducationalDataMining.org", 2010, `http://www.educationaldatamining.org/`.

The main objective of EDM is to develop new tools for discovering relevant rules or patterns in the raw data. When the attention switches to the large scale applications of these techniques, some researchers call it Learning Analytics, indicating that it is more geared towards the broad applicability of rules and findings retrieved inside the field (Bienkowski, Feng, and Means 2012). This is important in order to be able to extend the results of a single or a small number of experiments that confirm the hypothesis into courses or institutions other than the ones considered and directly analysed.

Nevertheless, other researchers[1] pose the distinction between EDM and LA more on the methods used for analysis. In their view, LA is more general, as it also takes into account qualitative methods and human judgment – such as sentiment, influence and discourse analysis, sense-making model, and Social Network Analysis – whereas EDM seems to be only interested in relationship of the quantitative data about the educational experience.

Combining EDM and LA it is possible to support tasks for defining learner profiles, tracking behaviors and finding relevant dimensions to classify and interpret online user activities in TEL experiences. The main object of the researches developed in this field is to predict a model to measure the student performance with the idea to recommend improvements to the current educational practices. Two tasks currently classified inside EDM are particularly relevant to the present research: "Causal data mining" and "Distillation of data for human judgment" (Baker 2010), which respectively focus on the elicitation of the generating causal relationship for some of the rules found and on the distillation of higher level knowledge from a huge amount of information, in order to better support the human capabilities of judgment and decision making.

Based on a recent report elaborated by Bienkowski, Feng, and Means (2012) for the *U.S. Department of Education - Office of Educational Technology* called *"Enhancing Teaching and Learning Through Educational Data Mining and Learning Analytics"* it is possible to identify some directions of research in which these approaches could help TEL to offer better experiences and provide educational results more in line with the requirements of modern education. The information and knowledge extracted could be re-framed to different time–scales and devoted to distinct roles, as indicated in Table 2.1.

An aspect normally neglected or underundestimated is the IT costs associated with the application of these techniques, which are both economic and organisational. In fact, the application of EDM requires the storage of vast amount of data for the relevant time-frame in

[1] `http://users.wpi.edu/%7Ersbaker/LAKs%20reformatting%20v2.pdf`

Role	TimeFrame	Scopes
Learner	immediate real-time	– selection of the next problem – feedback on subject completed – strong and weak/deficient personal knowledge
	weekly	– improvements in the last week – strong and weak personal knowledge areas
	semester	– improvements in the last semester – courses passed and not passed – suggestion for the next semester plan
Tutor	some hours	– monitoring of learner activities – near–immediate scaffolding intervention – providing feedback on the current activities
Teacher	daily	– next day's teaching adaptation
	weekly	– didactic plan advancement
Teacher coordinator	monthly semester	– judging educational progress – realigning the didactic load – identifying possible difficulties in learners
School Administrator	yearly	– overall school improvements – adaptations for the next school year – identification of best and problematic cases

Table 2.1: Possible application of EDM and LA, distinguished by the objectives, the optimal time-frame and the role interested in. Adapted from Bienkowski, Feng, and Means (2012) 'Enhancing Teaching and Learning through Educational Data Mining and Learning Analytics'.

a quick and reliable fashion, to guarantee prompt access but also to respond to the need to offer continuous education for the heterogeneous, yet very specialised, professionals that it will support and to improve and validate the algorithms underpinning the procedures. Not to be neglected are the security and privacy constraints to be respected alongside with the ethical obligations related to treating student data.

For an effective and fully meaningful usage of EDM and LA, the authors of the report suggest some basic directions to be followed:

- *Cultural change* => Using data for making instructional decision is a process that requires time and effort and which has to be supported to help Teachers and Instructional Designers to understand it and make the best of it
- *Consider IT* => Involve the IT departments in the design phase of the educational experiences as well. Use the suggestions they give to structure the experiments in the best way possible, also with regards to the collection and further reuse of the data of interest
- *Information Usage* => Support all the user of the information provided (visual or graphical, if possible) so that they become *smart data consumer* i.e. able to explore the information and to obtain the most useful knowledge
- *Pilot* => Start with pilot areas where the support of these tools seems more promising and concentrate the effort on those. Afterwards, progressively extend the successful cases to cover broader areas
- *Communicate* => Involve students (and even parents, if the case, such as in compulsory educations like K-12) reporting to them where, when and how the data is captured and how it will be used
- *Conform* => Try to conform to already existing standards and reuse well–known approaches when feasible, always respecting the technical limitations and the policies regulating the treatment of the data in the specific institution, environment or state

The authors of this report (Bienkowski, Feng, and Means) also indicate some research directions, whose results can guarantee a future real adoption and positive impact of EDM and LA in the education sector as one of the pillars of its innovation:

- *Usability* => Improve the usability aspect of the tool's design and interface that provide information to the users

- *Effectiveness* => Monitor the effectiveness of the tool, both with internal (i.e didactic results) and external (i.e satisfaction and willingness of use) drivers
- *DSS* => Use the extracted information to develop tools able to support the human judgment and decision process (DSS)
- *Extend* => Understand how it would be possible to extend predictive models already elaborated in domains/contexts other of the current one

The development of GVIS – and of the semantic data layers (i.e: the formalized description of sources, operations and encoding steps to be used, that will be explained in details in the next chapter) – tried to respect these indications and quality measures as much as possible. Despite these analogies, EDM (and LA) approaches are normally based on computationally intensive and fully automated data analysis, where GVIS is mainly based on the application of semantic layers of data that the Instructional Designers provide. This means that the main difference is the presence of layers that give an interpretation of the data retrieved as well as of the aggregation operations (semantic approach), upon which the process can rely to extract interesting information.

This rather different approach provides a way to guarantee a meaningful didactic interpretation of this data without the need for a further extensive and time–consuming validation phase as would be the case of a pure EDM approach. Obviously, the definition of the semantic layers for data extraction, identification, validation, fusion, distillation and representation is a quite challenging task for Instructional Designers. It requires in fact a good level of experience, a clear pre-identification of objectives and constraints and a critical thinking approach.

Nevertheless, thanks to these challenging tasks, the results automatically comply, for the most part, with the must-have objectives of a supportive educational tool i.e. 'Analysis and visualization of data', 'Providing feedback for supporting instructors' and 'Detecting undesirable student behaviors', as defined in the work of (Romero and Ventura 2010).

2.5 Possible impacts of learner model externalization

As already stated, Intelligent Tutoring Systems and Adaptive Educational Hypermedia have a built-in component of the student modeling procedure that maintains a representation of the learner's knowledge based on the detailed monitoring of the students' behavior within the system.

Traditional learner models were hidden from students and used exclusively by the system in a opaque way to adapt its behavior to individual users. However, recent studies in student modeling argued in favor of Open Learner Models. Bull and Kay (2007) pointed out the key purpose of presenting the model to students is to support meta-cognitive activities such as reflection, planning and self-assessment by providing feedback in respect of the students' learning and knowledge.

Moreover, it is possible to extend the student model with information about their peers. This type of model is called Open Social Learner Model (OSLM) (Hsiao, Bakalov, Brusilovsky, and Konig-Ries 2011) and benefits from both meta-cognitive and social aspects of the learning. In literature, analysis that concentrate or scrutinise the foreseen, or found, impact of OSLM begin to emerge, such as in (Falakmasir, Hsiao, Mazzola, Grant, and Brusilovsky 2012).

2.5.1 Adaptivity

Many researches have already explored the impact of OLM, both in supporting self-reflective processes of learners and in empowering the teacher on supervising tasks. Examples of related findings can be found in the works of McCalla (Bull, Greer and McCalla 2003), (Looi, McCalla, Bredeweg, and Breuker 2005), and (Tang and McCalla 2004); Kay (Kay and Kummerfeld 2010), (Kay 2008), (Kay, Reimann, and Yacef 2007), and (Kay 2006); and Bull (Ahmad and Bull 2009), (Shahrour and Bull 2008), (Bull, Gardner, Ahmad, Ting, and Clarke 2009), (Bull, Dimitrova, and McCalla 2007a), (Bull, Dimitrova, and McCalla 2007b), and (Bull and Kay 2008). Despite this massive attention to the theme, no previous work has been specifically concentrated on the representational aspect of OLM. Rather, each work reported using its own fixed, well-known, and established paradigm for the visualisation of the learner model information (whether a network-based, a bar-chart derived, an iconic metaphor or a textual representation).

A remarkable exception is (Mabbott and Bull 2004), where a study is presented in which individual learner models were presented to student, allowing them to choose which view to adopt. The learners were able to appreciate this functionality and take full advantage of it. This assumes a high relevance when interaction with the OLM is individualised.

As a result of this finding in (Mabbott and Bull 2004), there is the need to further explore the theme of personalisation in the information presentation. A further step in this direction is the adoption of an adaptive OLM representation, where the model can be automatically adapted to the learner based on information in the profile, the contextual information of the user task currently in progress or the general preferences declared.

A foreseen impact of an adaptive representation could be the limitation of the cognitive overload[1] through the filtration of the amount and complexity of the model presented based on the current cognitive load as well as the experience of the learner with the system. We expect that this aspect can be measured indirectly with structured questions about the learner experience and the perceived usefulness of such a tool. This is an open research question.

2.5.2 Social aspects

The exploration of the effects of social information representation for didactic purposes is a relatively new theme as found in (Hsiao and Brusilovsky 2012). Some previous research Glahn (2009) and Kehrwald (2010) stated that is important to increase the learner social presence inside a LMS or a Virtual Learning Environment (VLE). The most common proposal found is to replicate the native capabilities of real-world environments inside the learning environment, such as offering functionalities to learner to compare himself with the class average or to find other peers who are similar or complementary in terms of strengths and weaknesses.

Study results demonstrated a higher level of engagement by the learner when social capabilities are included in the system. In fact, they tend to spend more time working with the materials - consulting resources, answering self-assessment questions and working on available exercises - and, as a result, they seem to achieve higher success rates in comparison to those without social capabilities.

2.5.3 Big, Heterogeneous and Distribute data sources

Some other researches are also relevant from the point of view of information treatment, merging and usage, struggling with complex and/or extensive data sets that need to be represented, as in the work of Mazza and Dimitrova (2004) , Van Labeke, Brna and Morales (2007) ,and Zapata-Rivera and Greer (2004). From this point of view these studies expected to provide a contribution showing that a general purpose tool could be an option to provide a uniform externalised representation of the learner model from different and heterogeneous learning platform sources. A theme that has not received sufficient coverage yet is around the application of profiling methodologies in the context of Life–Long Learning experiences, where the user data is distributed in different and frequently not connected learning platforms or environments (Heckmann, Schwartz, Brandherm, and Kroner 2005).

[1]this term, that derives form the *Information Overload* concept, indicates the additional effort required by the users in order to make sense and effectively use the additional information provided by the OLM, which is usually different from the learning task assigned to him/her.

In the context of the GRAPPLE project, an analysis of this aspect was performed and, as a result, a facility was developed which is called GRAPPLE User Modeling Framework (GUMF). This facility is in charge of receiving data from the sources (here they are represented by different learning platforms, such as Sakai, Moodle, Claroline, IMS-CLIX and Elex) and create a unified user profile, as described in some of the project publications (Abel, Henze, Herder, and Krause 2010), (Leonardi, Abel, Heckmann, Herder, Hidders, and Houben 2010), and (Abel, Herder, Houben, Henze, and Krause 2010).

During the development of the GVIS tool a choice was made to maintain a certain neutrality with respect to this theme and as a result the software has the capability to connect simultaneously to distributed and heterogeneous data sources. This means that the same operation *extraction*, *aggregation* and *visualisation* could rely on different data sources. Unfortunately, it is very difficult to find didactically-related data sources that could be connected seamlessly, mainly due to the following two problems. First of all is the issue of finding an unique user identification across different systems, and secondly are the privacy and security issues raised by the transmission and usage of the data outside of the platform that generated it.

In GRAPPLE we tried to tackle both these problems through the creation of a distributed architecture with federation capabilities (i.e. able to provide centralised, global identificators) and the adoption of a common broker mechanism to deal with the management of basic privacy policies.

2.6 GRAPPLE: context of the work

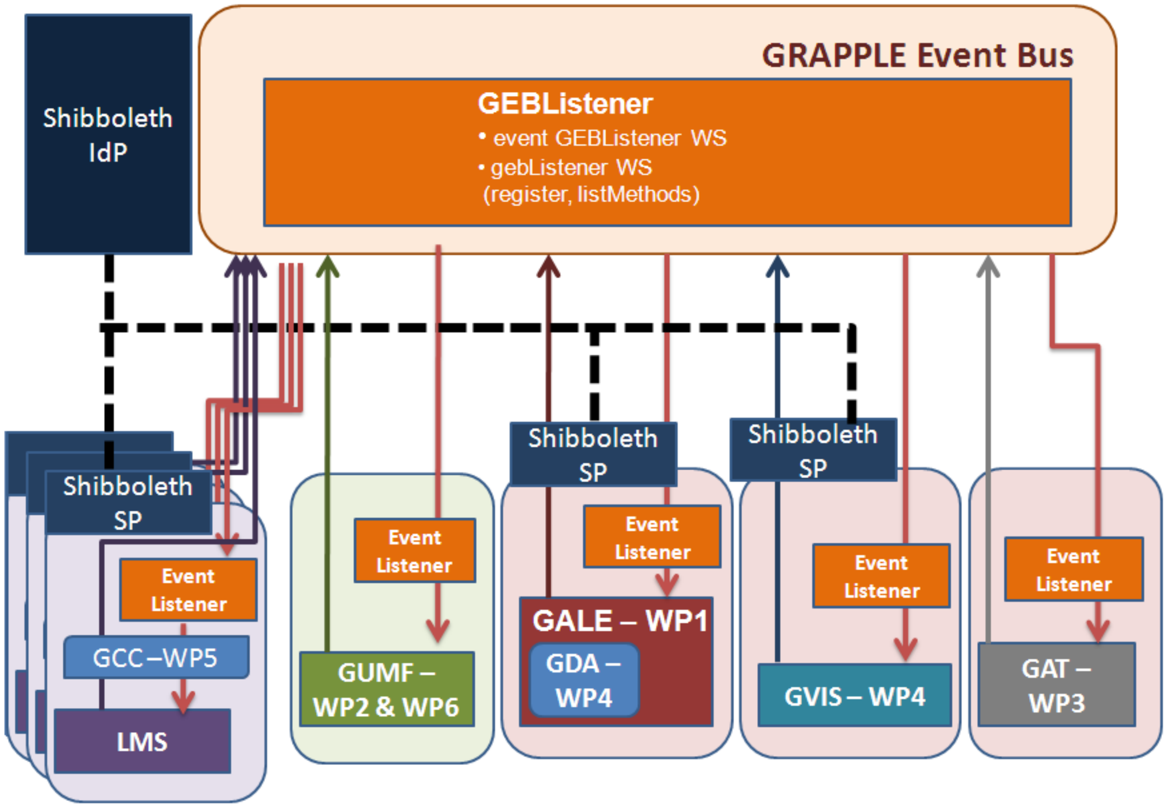

Figure 2.1: The GRAPPLE General Architecture

The GRAPPLE Project (Generic Responsive Adaptive Personalized Learning Environment) aims at delivering a TEL environment to learners that guides them through a learning experience, automatically adapting to personal preferences, prior knowledge, skills and competences, learning goals as well as the personal or social context in which the learning takes place.

The system includes a user model infrastructure that keeps track of the learner's knowledge and skills acquired during the learning process. This knowledge is available to the learners and instructors by means of interactive visualisations that can be performed in GRAPPLE or seamlessly included in any other tool that participate in the Personal Learning Environment.

The visualisations take into consideration the learner model, the domain model, and the adaptation model inside GRAPPLE, but it can also include other data coming from distributed and heterogeneous sources. The project has two distinct objectives: to directly support the explorative and self-reflective process of the learner; to help instructors assist the learners. One of the upsides of this set of visualisations is that they can support learners to be more engaged in the learning process.

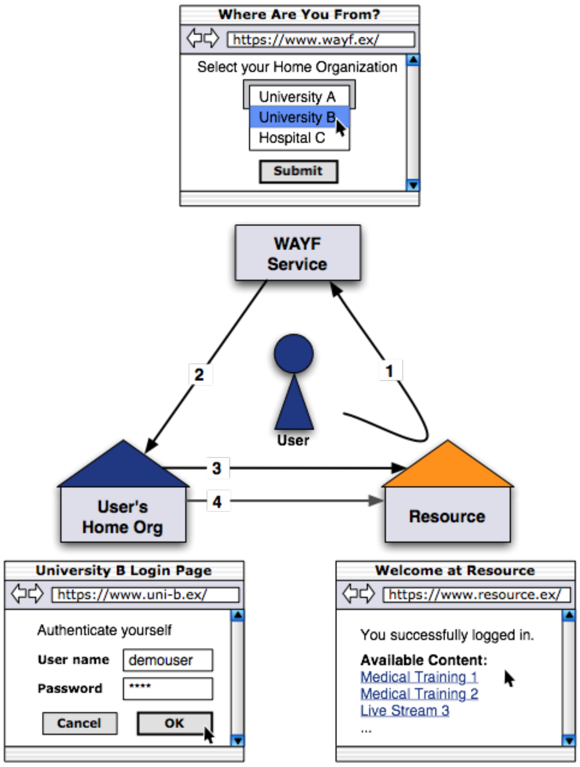

Figure 2.2: How Shibboleth works. Images from `http://www.switch.ch`

This section lists a number of outcomes of GRAPPLE project, specifically in relation to the GVIS part[1].

The study in the context of GRAPPLE was organized in different phases, each one of them concluded by the respective document describing the findings and reached objectives. In its first phase of the study the state of the art in the field of user modeling was analysed. To follow was a requirements analysis, undertaken with potential users of the systems through a series of meetings and interviews. An initial description of user model (UM) and domain model (DM) components (which provided the most relevant input data for the visualisations) were produced based on the above feeds. An analysis of the possible scenarios of usage was conducted which

[1]further information about GVIS and the rest of the infrastructure can be found in the public deliverables section `http://grapple-project.org/public-files/deliverables/` on the grapple website.

would provide a guideline for the following phases of the work. Last but not least, the technical infrastructure was designed and engineered.

In order to achieve an operational state, the description of the available and useful sets of data extracted from other GRAPPLE components was defined, the aggregation technique with the current data set was sketched, the initial visualisations designed and the architecture of the software created. In last part of the project, the architecture and implementation of the module was finalised. This was followed by an analysis of the tool, which required the involvement of the final users. This final step was divided in two phases: an evaluation of the planned visualisation with the aid of mock-ups, and a different evaluation carried out with the GVIS integrated in the GRAPPLE infrastructure, as would be the case in a real, adaptive course.

The infrastructure (see Fig. 2.1) comprises the implementation of several components which are able to support the inclusion of adaptive contents in existing LMSes (here represented as a set of light purple rounded block, on the left).

Their seamlessly inclusion is achieved through the GRAPPLE Conversion Component (GCC) that makes contents stored using the GRAPPLE internal format compatible with the format of others LMS. The GCC also provides the possibility for the LMSes to share controlled sets of data about their users which are stored in the central GRAPPLE repository GUMF.

As described in Fig. 2.2, the authentication is managed at federated level by a Shibboleth supported Single SignOn (SSO)[1], which supports an Identity Provider (IdP) for every partner participating in the federation and one SP (Service Provider) for every service registered.

In GRAPPLE, the course contents are built with respect to conceptual domain and adaptation structures, based on concepts. The domain model is based on the concepts and their relationships, which are enriched by the contents that represent the existing concepts and their relative links. The user model stores, amongst other things, the concepts list and the related knowledge level, which is a measure of the learner's knowledge on a specific concept.

Putting together the different models stored in the system – the Domain Model (DM), the Content Model (CM), and the User Model (UM) – is possible to support the generation of the actual adapted content to be provided to the user in the specific course.

With the adoption of this process, every concept could be presented to the user using one of the contents (i.e. a web page, a video or whatever could be represented as an URL) related to it in the model provided by the instructional designer. The conceptual adaptation model

[1]Single SignOn is an approach to provide user authentication using a dedicated service, which can be seamlessly connected with to arbitrary number of systems to manage user authentication on their behalf.

includes the adaptation policies and rules, the expected knowledge levels for the concepts and the expected didactic goals. The GAT (GRAPPLE Authoring Environment) is the component that allows instructional designers and teachers to author courses in the form of concepts of the Domain Model and adaptation rules. The GALE (GRAPPLE Adaptive Learning Environment) is the run-time engine that provides the adapted content to the learners.

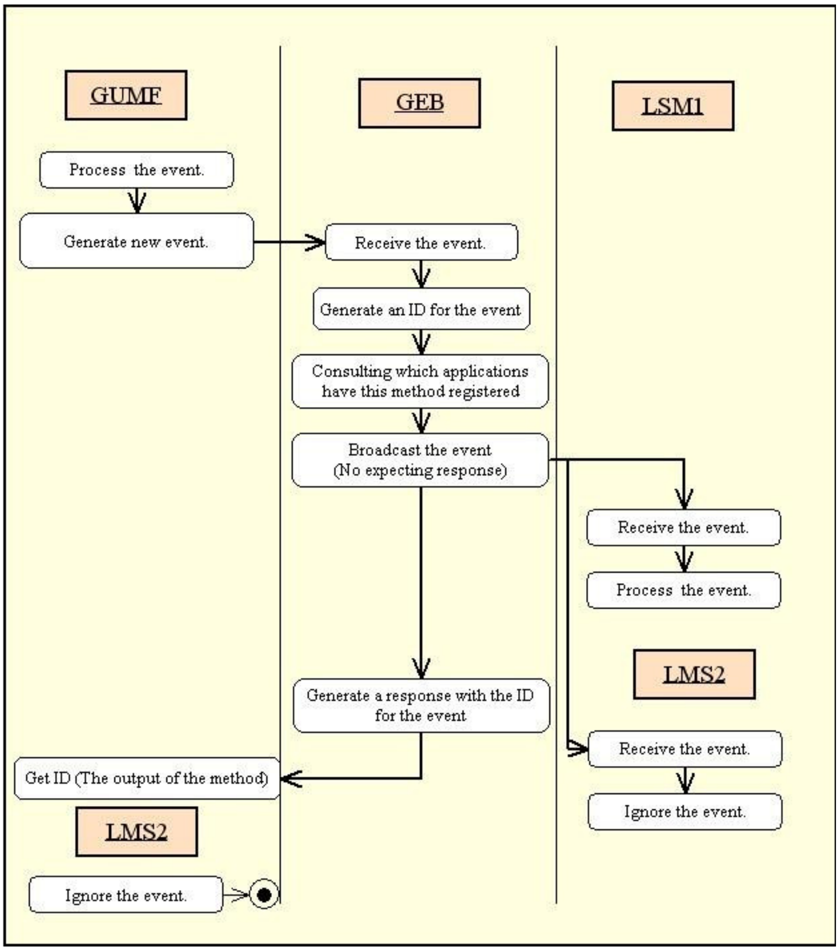

Figure 2.3: The mechanism of notification adopted in GRAPPLE. GUMF generates a new event either based on information received directly or by deriving new knowledge through a reasoning rule; GEB receives an event and broadcasts it to all the subscribed component, which can decide to ignore it (such as LMS2 in this case) or process it (LMS1). Hidden in the image are also the Grapple Listener and Broker.

The learning environments are able to seamlessly include the GALE user interface, as showed in one of the deliverables[1]. The adapted course is included in the educational platform similarly

[1]the deliverable D7.1c (Mazzetti, Tenerini, Dicerto, Van der Sluijs, Smits, Rambout, Abel, Pekczynski,

to any other learning activity.

All the data about learners coming from the educational systems and the GALE facility is stored through the Grapple Event Bus (GEB) in the GRAPPLE User Modeling Framework (GUMF), a distributed user modeling facility that enforces the correct interpretation of the data received and the application of reasoning rules. The notification of availability of updated information is made possible through a broker-listener mechanism managed by autonomous subscriptions and by single component, as showed in Fig. 2.3. One of the component involved in collecting the circulating data is the Grapple visualisation tool (GVIS), even though it relies mainly on data directly stored in GUMF and queried in RDF format. Nevertheless, since some other learner data is not directly stored in the internal user modeling framework – and also to remain open to further integration –, GVIS was designed to collect and aggregate data from other sources that are not integrated in the GRAPPLE framework, such as data from social web services, intranet usage data, web navigation footprints and so on.

Mazza, Mazzola, Gaeremynck, Foss, Minne, and Vasilyeva 2011) - Final specification of the operational infrastructure, publicly available on the project website (`http://grapple-project.org/public-files/deliverables/GRAPPLE-D7.2c-Data%20models-v1.0.pdf`)

Chapter 3

GVIS: the tool

This chapter is about the requirement elicitation, ideation, and development of the component in charge of offering the possibility to create adaptive visual representation of the learner model. In particular, the general structure of the developed infrastructure is presented and a basic explanation of the configuration XML[1] files that encode the semantics of the operations on the data is provided. Furthermore, a short section about the development process is reported for completeness.

3.1 The Infrastructure

GVIS – acronym for GRAPPLE Visualization Infrastructure Service – is a module developed as part of the project to extract data from different sources and enable instructional designers to easily create adaptive indicators of the learning state for learners, tutors, and teachers.

3.1.1 Concepts

The GVIS module is a three-tier software architecture[2] that allows for great flexibility, customisation and independence of the source data coming from other system components. It has been designed to provide the following features:

- Possibility to extract data from different data sources and using different methods (web services, database queries, Semantic Web (SPARQL) and others)
- Possibility to change or extend the set of source data without having to change the software

[1] *XML* stands for eXtensible Markup Language and is one of the well–known and affirmed language for structured data encode and exchange over Internet services.

[2] A three-tier architecture normally separates the data level from its semantic and its presentation.

- Possibility to define, amend or extend data aggregations without the need to rewrite (or change) the software
- Possibility to include adaptation on the graphical representations through the inclusion of programmable conditions in the configuration files

This flexibility is achieved through a design that uses XML configuration files to encode all parameters for data extraction, to define operations on data as well as data aggregations and to specify adaptation rules on graphical representations, as outlined in Fig. 3.1.

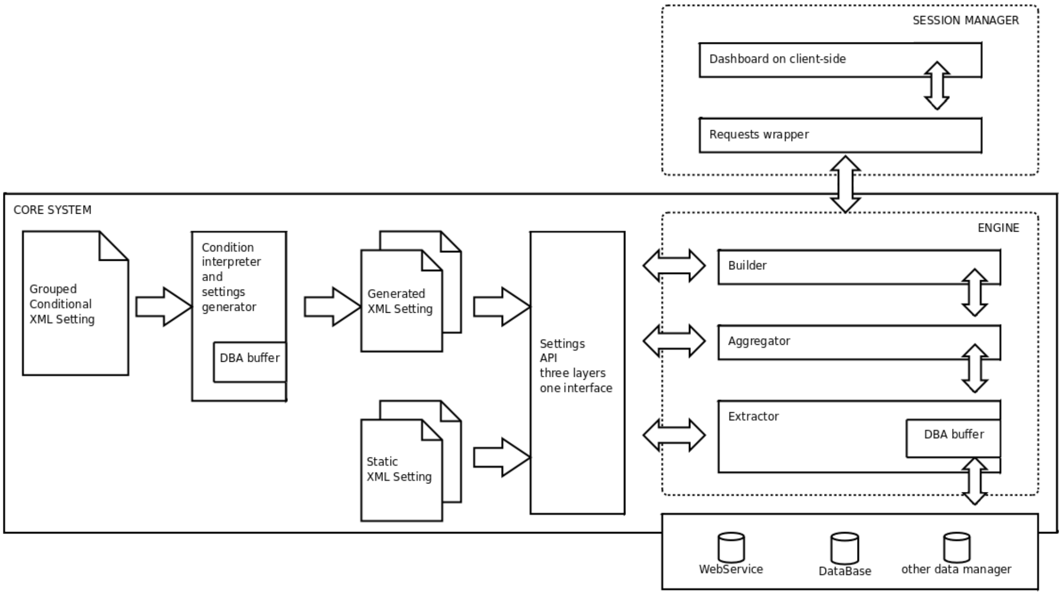

Figure 3.1: The GVIS architecture with the three levels highlighted inside the Engine block: Extractor, Aggregator and Builder.

3.1.2 Requirements and Design

Data about user activities and status is stored in the form of logs and tracking data inside the LMS to provide a first elementary user profile and activity report (regardless of whether other systems already memorised it in the learner model, at a higher level of aggregation and abstraction). The user profile is then created on the basis of their activities and interactions in the learning environment. Although many Learning Management Systems already provide the possibility to explore this user tracking data, in many cases the visual presentation of the information is not well suited to the users' specific needs. The exposed data is generally provided as a simple list of user activities – or access to course contents – without the possibility to explore it from an aggregated didactic point of view. For this reason, adding a didactic oriented view requires an effort that normally requires interactions between different skills, to cope the objectives of the activities with the technical way of implementing it and interpreting its logs. This means that the didactic interpretation of the logs, i.e. the meaning to be given to this data in the educational experience, has to be added by the ID/teacher (in cooperation with the technicians) each and every time. In case where the learning experience was developed in a team oriented flavor (teacher, Instructional Designer, tutor, technician are different persons) this effort can become demanding if multiple refinement recursions are required to reach the expected results and carries the risk of generating inconsistencies in successive improvement iterations. In fact, this events exploration feature was originally thought for technicians in charge of solving technical issues, rather than for instructors or tutors interested in improving pedagogical aspects.

In other works, the presentation of the data is either limited to a data subset or predefined by developers and therefore fixed (Mazza and Milani 2004), (Mazza and Milani 2005), and (Mazzola, Eynard, and Mazza 2010). Notable exceptions in the field of OLM are OLMlets (Bull, Gardner, Ahmad, Ting, and Clarke 2009), in which the learner can choose autonomously between seven different representations. Nevertheless they still rely on data coming from a single system, normally the one on top of which they were developed.

GVIS provides an easy way to create effective graphical presentations of arbitrary data from different and heterogeneous sources through the three-tier architecture consisting of a data extractor, a data aggregator and a builder, as shown in Fig. 3.1.

All these levels rely on a configuration file that the instructional designer can amend or expand to add graphical indicators (in the form of widgets) of one or more interesting characteristics of the user profile. The infrastructure can connect to any data source with different connection types (e.g. databases, Web services, connection bus) simply through the creation of a small adapter. The output produced by the tool, as seen by a final user, is a flash based interface that represents, using one of the available graphical metaphors, the information that is considered relevant for the user itself aggregated according to a didactic model, which is the interpretation of elementary actions such as opening a web-page, posting on as forum or answering a quiz.

In order to achieve this result, it relies on a highly configurable infrastructure based on layers, each one implementing a level in the model: extraction, aggregation and widget creation.

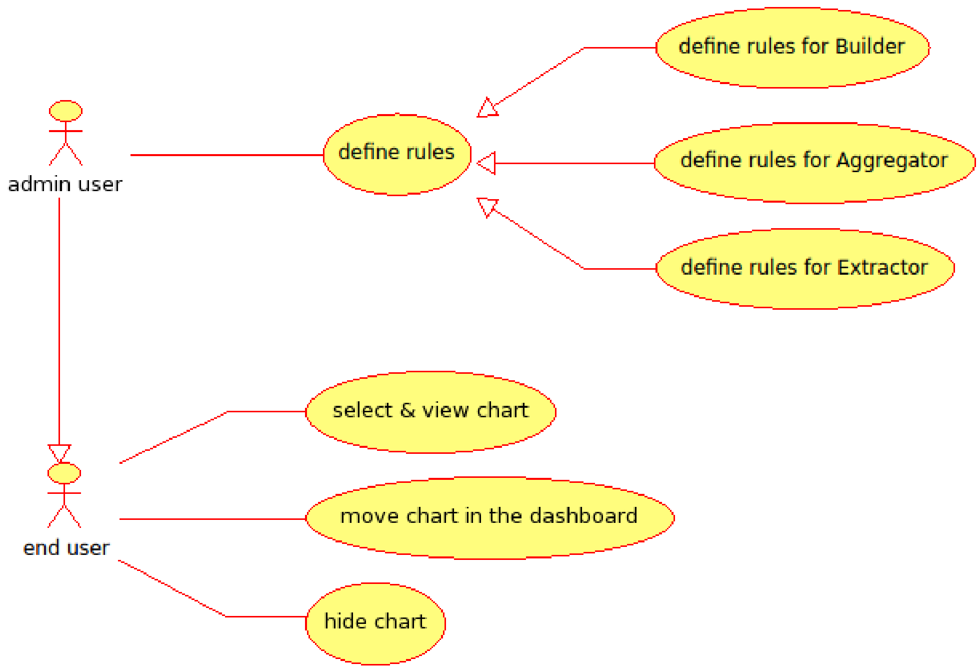

Figure 3.2: The GVIS requirements for the general user.

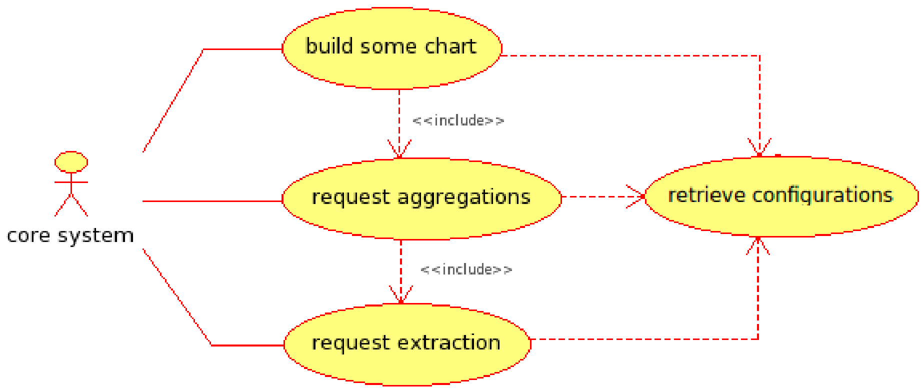

Figure 3.3: The GVIS requirements for the server-related part.

Furthermore, this schema follows a common data processing pattern: retrieve raw data, extract or derive, and present it in the most suitable way (Mazzola and Mazza 2009a), (Mazzola and Mazza 2009b). This approach allows the tool to create a sort of educational mash-up[1].

[1]It is called *mash-up* the software or service that relies mainly on different and not naturally interconnected sources for providing a higher value service, based on the integration and enrichments of the single data source adopted; i.e. the fusion of awareness data with representational facility (map), such as in the case of data

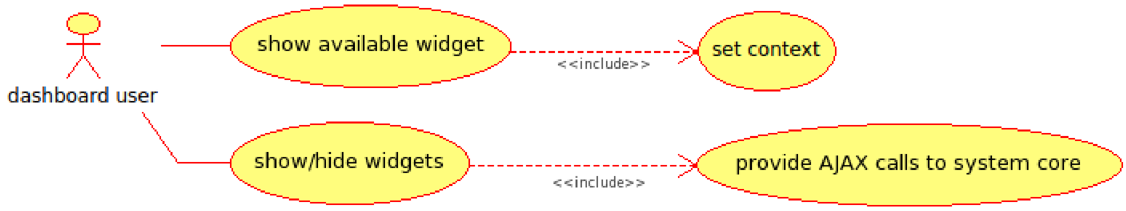

Figure 3.4: The GVIS requirements for the browser-related part.

The requirements are presented in the UML schema included in this section. Fig. 3.2 shows the basics functionalities: the *end user* can perform three operations (*select and view chart*, *move chart in the dashboard*, and *hide chart*), whereas an *admin user* can work with the configuration, defining rules for the three levels of the tool. Figure 3.3 presents the operational view, as seen by the *core system*, offering the capability for request extractions, aggregations and chart creation based on requests received from an *end user*. On the other side, Fig. 3.4 offers a view of the *dashboard* one, which provides a list of all the available widgets and allows to change their visibility status in order to support the *final user* interactions.

Finally, Fig. 3.5 represents the logical flow of information for the widget generation phase. It begins with a request (formulated by the user in the dashboard interface and encoded in an AJAX[1]) request that is intercepted by the core-system to invoke the relevant Builder. The Builder object then takes the control and checks the configuration which, if successfully validated, invokes the set of Aggregators required one at a time. Each individual Aggregator requests the set of Extractors it requires. Eventually the data is turned into information through composition and computation and is returned to the builder that, in the final step, maps it to the chart and returns the newly created object to the dashboard.

regarding current locations of a list of friends provided in textual way, enriched by its representation on a map with distance (Km) and time required to reach them by car given the current state of the traffic in the area

[1]The acronym AJAX stands for Asynchronous JavaScript And XML and is a *de-facto* standard for the request of a segment of information to be inserted in a answer or in a segment of a web-page once the page is already loaded in the client browser. It works through the capabilities of the JavaScript implementation in the browser to react to asynchronous events and to manipulate the Data Object Model (DOM) of the document rendered and the method HttpRequest for requesting an informative element through the normal http connection.

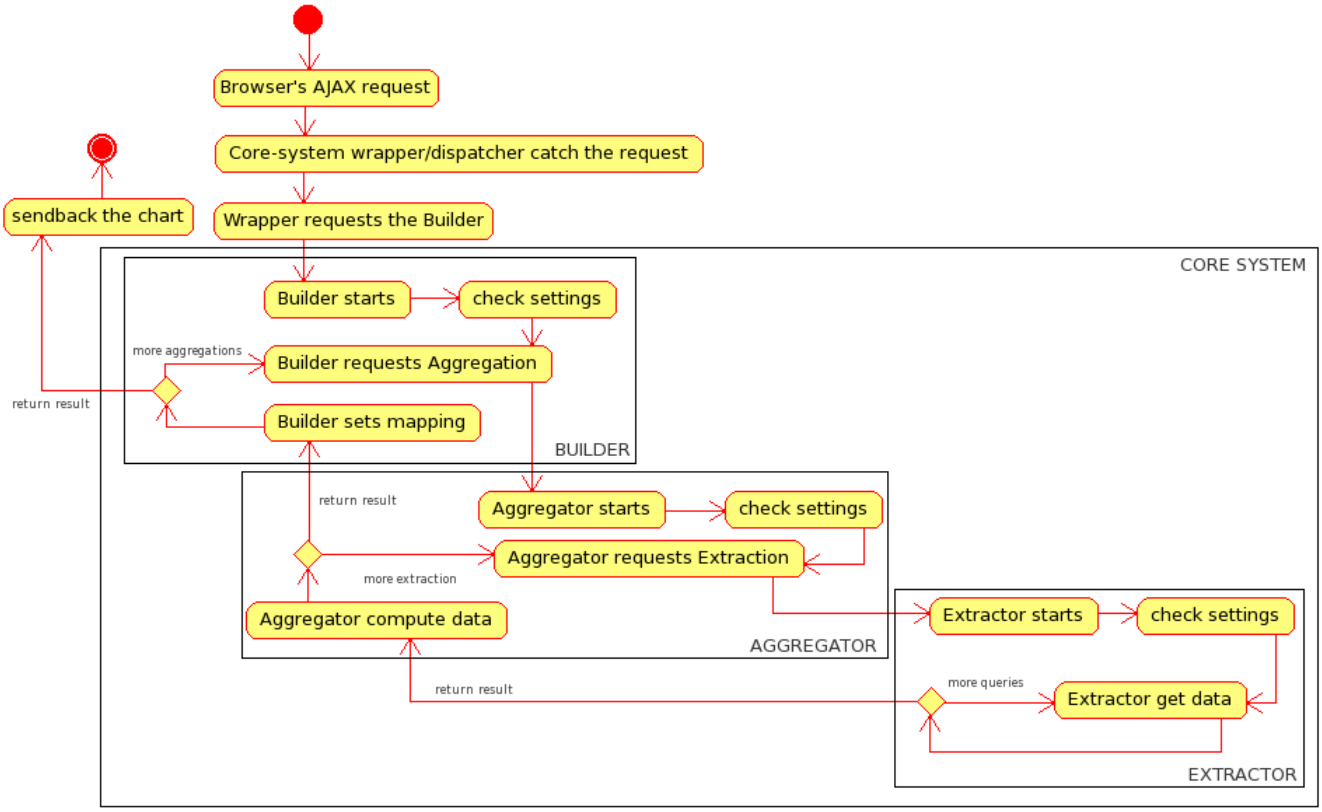

Figure 3.5: Data flow in the widget generation.

3.1.3 Mash-up of data in web 2.0

Because Web-services and RDF[1] data model play a major role in integrating distributed services for Web 2.0, we developed an infrastructure with the ability to take into account different types of data sources by means of configuration profiles. The process of including facilities in a liquid and adaptable environment requires not only the availability of a standardised way to namelessly connect the active component to the environment that will host it (known as the container), but also an effective method – like the JSON [2] data format – to exchange data between different applications, services and data storage facilities. Fig. 3.1 shows the current infrastructure of GVIS in a comprehensive fashion and offers further details on the connection with heterogeneous sources. The system is flexible as the behavior of its main components can

[1] RDF stands for Resource Description Framework and is a way to represents facts using a triplet i.e. a subject, an attribute and a value. It is well suited for data merging even if the underlying schema differ.

[2] JSON stands for JavaScript Object Notation, a string–based and Internet transparent notation for encoding object to be transmitted through the header or payload of the page

be easily changed by modifying the related XML configuration files.

Although the tool is designed for usage in the context of a distributed and heterogeneous environment, for the purpose of a first test case this architecture was applied to a single data source (represented by a course deployed on an institutional LMS). Even if this case does not present distributed characteristics, it is important to consider the fact that the single source of data is external to the tool. On the other aspect, the mash-up of data, it has to be noted that this requires more than one source that are naturally related to the learning experience which are not easy to find, except in cases where the educational experience was conceived to be supported by different tools from the very beginning. As a proof of concept, in the context of GRAPPLE, the same tool was used in conjunction with a federation of different learning platforms. In order to test the practical feasibility of mashing up data, other experiences were designed and implemented that rely on completely different sources of data, like folksonomies and tracking logs from personal browsing history (Mazzola, Eynard, and Mazza 2010).

In the following subsections a description of every module shown in Fig. 3.1 is presented, both in term of functionalities provided and basics of the XML configurations.

3.1.4 The Extractor

The extractor represents the lowest level of our application and is in charge of retrieving data from the sources, as showed in the Fig. 3.6. This piece of software takes care of producing a syntactical and semantic translation of the data received from a particular source to the internal format. The semantical part can be used to translate from a measure system to another or to normalize the data. Both the pull and the push approaches can be implemented to retrieve and collect data. To achieve this objective they rely on a small amount of formal code that describes the data structure used by a particular source. The full schema of the configurations are presented in the Appendix A. The following excerpt of a configuration file is particularly useful to explain some peculiarities of the module:

```
[1]         <source name="MoodleEvaluationGlobal">
[1.1]       <accessinfo>
[1.1.1]     <accesstype>DB</accesstype>
[1.1.2]     <accesspoint>**IP**</accesspoint>
[1.1.3]     <accessmode>mysql</accessmode>
[1.1.4]     <accesssource>**DB_name**</accesssource>
[1.1.5]     <username>**UserID**</username>
[1.1.6]     <password>**PWD**</password>
```

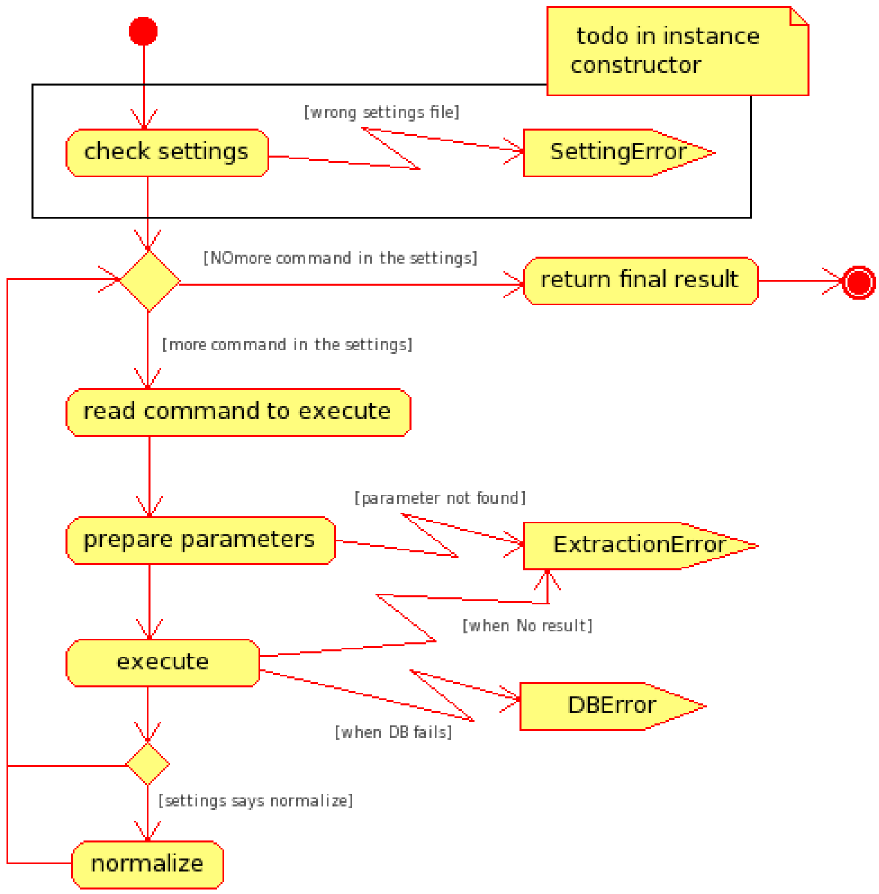

Figure 3.6: Actions involved in the *Extraction* execution step.

```
[1.1.7]     <lifetime>30</lifetime>
            </accessinfo>
[1.2]       <query>
[1.2.1]     <sql>
            select GI.userid AS name, GG.finalgrade AS value
            FROM mdl_grade_items AS GI JOIN mdl_grade_grades AS GG
            ON GI.id=GG.itemid WHERE ... AND GI.courseid=?
            ORDER BY finalgrade DESC
            </sql>
[1.2.2]     <parameters>
[1.2.2.1]   <param>course.id</param>
            </parameters>
[1.2.3]     <resulttype>listofrecords</resulttype>
            </query>
```

```
            ...
            </source>
[2]         <source name="MoodleEvaluationSingle">
[2.1]       <accessinfo>
            ...
[2.1.7]     <lifetime>0</lifetime>
            </accessinfo>
[2.2]       <query>
[2.2.1]     <sql>
            ...
            </sql>
[2.2.2]     <parameters>
[2.2.2.1]   <param>course.id</param>
[2.2.2.2]   <param>user.id</param>
            </parameters>
[2.2.3]     <resulttype>numeric</resulttype>
            </query>
            ...
            </source>
```

In section *[1.1]* all the parameters for the connection with the data source are included, in *[1.1.1]* the type of adapter class is declared, together with *[1.1.3]*, which refines the previous indication. Section *[1.1.7]* defines the buffer lifetime for the extracted information: in the specific case the value of 30 means that the system will buffer and reuse the data for all the following requests that will occur within a time-frame of 30 secs. This could be useful for data sources having a slow response time. If this functionality is not needed, a 0 value can be used like in fragment *[2.1.7]*. In the second half of the source configuration a specific query is inserted (like in *[1.2.1]*), with one or more parameters (see *[1.2.2.1]* and *[2.2.2.1]*, *[2.2.2.2]*). In the final part is a declaration of the expected output type, whose range could be one of the following: ***numeric*** *[2.2.3]*, ***record***, ***list*** or ***listofrecords*** *[1.2.3]*.

Figure 3.6 specifies the flow of data extraction: an instantiation of the Extractor, after positively verifying its configuration (generating a *SettingError* exception otherwise), reads all the requests contained in it and executes each one of them, after having associated the actual parameters to the extraction step. If any unexpected or wrong result is returned to the instance, an exception is raised (respectively of the type *ExtractionError* and *DBError*). A final *normalization* step can be applied to the extracted data, if explicitly indicated in the configuration applied. Finally, if no exception is generated, the end result of the extraction is returned to the instance of the invoker at the upper level.

3.1.4.1 The Adapter

As already stated, the Extractor is able to connect to different data sources. This approach allows our solution to be seamlessly extended with different and heterogeneous data providers. When a new data provider is added to the infrastructure, a new mapping for the provider has to be provided as well. This can be done either by writing a new adapter class or reusing an existing one. An adapter is a class in the code that can be invoked by GVIS in order to deal with a specific type of data source. It uses the standard fields included in the configuration for retrieving the data, formatting it correctly and using the most suitable way to query the source. In the released version some specific classes are included for managing connections with MySQL (called DB, due to the fact that it can accept almost any source conforming to SQL[1] protocol), a specific Web-service interface and a SPARQL[2] endpoint interface.

3.1.5 The Aggregator

The aggregator is in charge of filtering raw data collected by the extractor and to apply some operations to the aggregated data, as clearly shown in Fig. 3.7. The instance starts with a validity and sanity check (with an exception *SettingError* which is generated in negative cases), then invoke the Extractions whose returned data is stored in order to apply the computations required and either return the results to the upper level or generate an *AggregationError* exception in case something went wrong.

This aggregation is based on the teaching and learning objectives that the teacher or instructional designer adopted, in fact it represents the information that is useful for learner and is strictly related to the pedagogical approach provided in the learning experience. This data aggregation into information – combined with its representation – could play an important role in supporting the learning process. With such an architecture we expect to offer a customised tool that can take into account different design models for didactic experiences. The use of models (based on XML syntax with an associated Name-space) provides a formal way for designing the behavior of the aggregator module.

[1] *Standard Query Language* is a specialised programming language for managing data in relational database management systems

[2] *Simple Protocol and RDF Query Language* is a specialised RDF query language

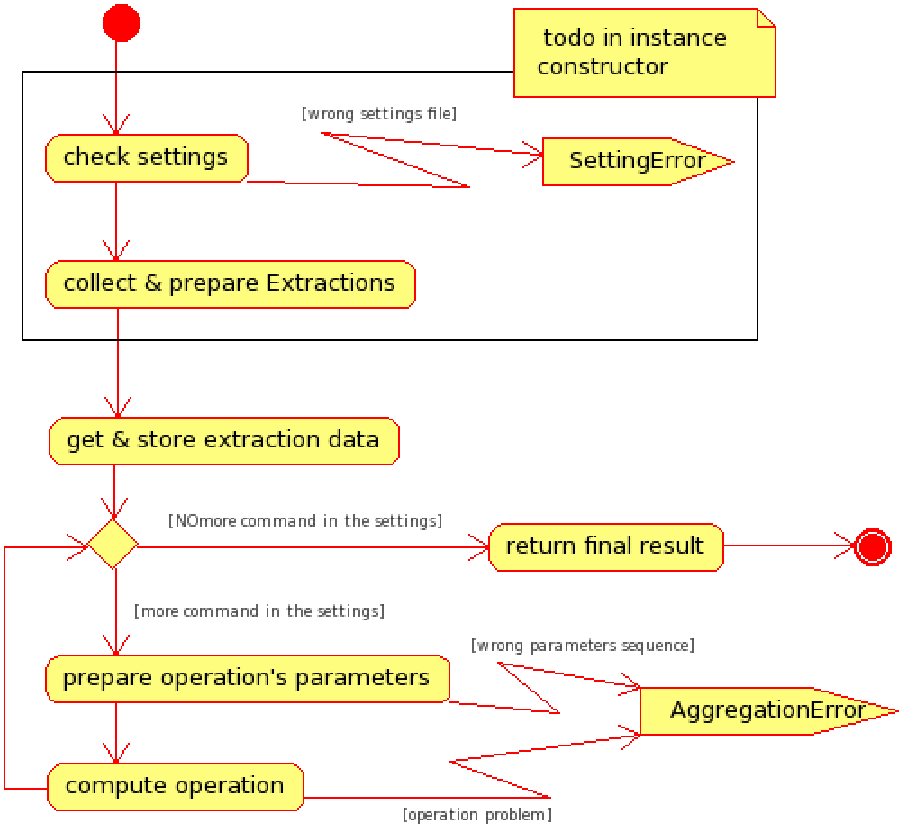

Figure 3.7: Actions involved in the *Aggregation* execution step.

3.1.5.1 Didactic models

Didactic models are defined by means of configuration files that describe how source data is aggregated in order to build meaningful and useful indicators. Here we show some XML fragments of this configuration that primarily describe which data is expected as input (like *[3.1.1]* and *[3.1.2]* or *[4.1.1]*) and which type of information will be produced as output (as in *[3.3]* or in *[4.2]*). The transformation process from input to output is also described in the form of a pipeline of operations: the output of a step could be used as input on a following one, like for the average operation in *[3.2.2.1]*, whose parameter compute is set to true (see *[3.2.2.2.1]*). Operations (like in *[3.2.1.1]* and *[3.2.2.1]*) are provided by internal classes that could be extended as required. The class name – like *ExtractCol* or *average* in the example – defines the operation performed and implements an abstract model which defines the expected properties and methods.

```
[3]          <source name="getMooEvalAvg">
[3.1]        <extraction>
[3.1.1]      <toextract>MEvalGlobal</toextract>
[3.1.2]      <toextract fix="true">1</toextract>
             </extraction>
[3.2]        <computation>
[3.2.1]      <tocompute>
[3.2.1.1]    <operation>ExtractCol</operation>
[3.2.1.2]    <parameters>
[3.2.1.2.1]  <param>0</param>
[3.2.1.2.2]  <param>1</param>
             </parameters>
 [3.2.1.3]   <resulttype>list</resulttype>
             </tocompute>
[3.2.2]      <tocompute>
[3.2.2.1]    <operation>average</operation>
[3.2.2.2]    <parameters>
[3.2.2.2.1]  <param computed="true">0</param>
             </parameters>
[3.2.2.3]    <resulttype>numeric</resulttype>
             </tocompute>
             </computation>
[3.3]        <resulttype>numeric</resulttype>
             </source>

[4]          <source name="getMooEvalSingle">
[4.1]        <extraction>
[4.1.1]      <toextract>MooEvalSingle</toextract>
[4.2]        </extraction>
             <resulttype>numeric</resulttype>
             </source>
```

3.1.6 The Builder module

The Visualisation module (a specialisation of the *Builder* module, specifically devoted to the representation of the data for the user) is the part that produces the actual visualisation. While this module has not been fully in its exporting capabilities implemented, other Building specialisation could be implemented, such as to format the data extracted for the purpose of connecting to other systems, for providing them the computed information.

The actual implementation – Visualisation – is divided in two components: the initial container, called dashboard, and the actual contents, represented by graphical widgets that map information into the final indicator in a graphical form.

The configuration of the dashboard can be personalised based on some parameters set at global level in the GVIS instance, like in the following XML fragment that defines the type (*[5.1]*) and data sources (*[5.1.1.1]* and *[5.2.1.1]*) to be used by the actual widget (such as the one represented in Fig. 3.12):

```
[5]<widget name="Note">
    ...
[5.1]  <chart type="hbar">
[5.1.1]    <chartsource>
[5.1.1.1]      <data>getMooEvalSingle</data>
               ...
           </chartsource>
[5.2.1]    <chartsource>
[2.2.1.1]      <data>getMooEvalAvg</data>
               ...
           </chartsource>
      </chart>
   </widget>
```

The generated widget is in the form of a horizontal bar-chart (*[5.1]*) and provides two types of information: the evaluation for the student (*[5.1.1.1]*) and its comparison with the class average (*[5.2.1.1]*).

3.1.6.1 The Dashboard and the Widget generator

The dashboard is instantiated once for each client –the browser of the user that connects to the platform for accessing the GVIS generated widgets– when the service is started. The actions and flow of information connected with the Builder invocation is presented in Fig. 3.8, where the *check settings* function verifies the configuration and creates all the instances of the *Aggregator* needed for the correct execution of the module *Builder*. The module provides two types of functionalities. It is a container for all the widgets and it collects all the user interactions and feedbacks, such as data filtering or widget visibility change. As a container, it provides a common place for the different widgets, each of which is specialised to represent a specific aspect; on the other side, by managing the interactions with the user, it acts as a sort of control panel of the learner situation. The interaction functionality is important because this is the only level at which the final user can express preferences or partially change the behavior of the

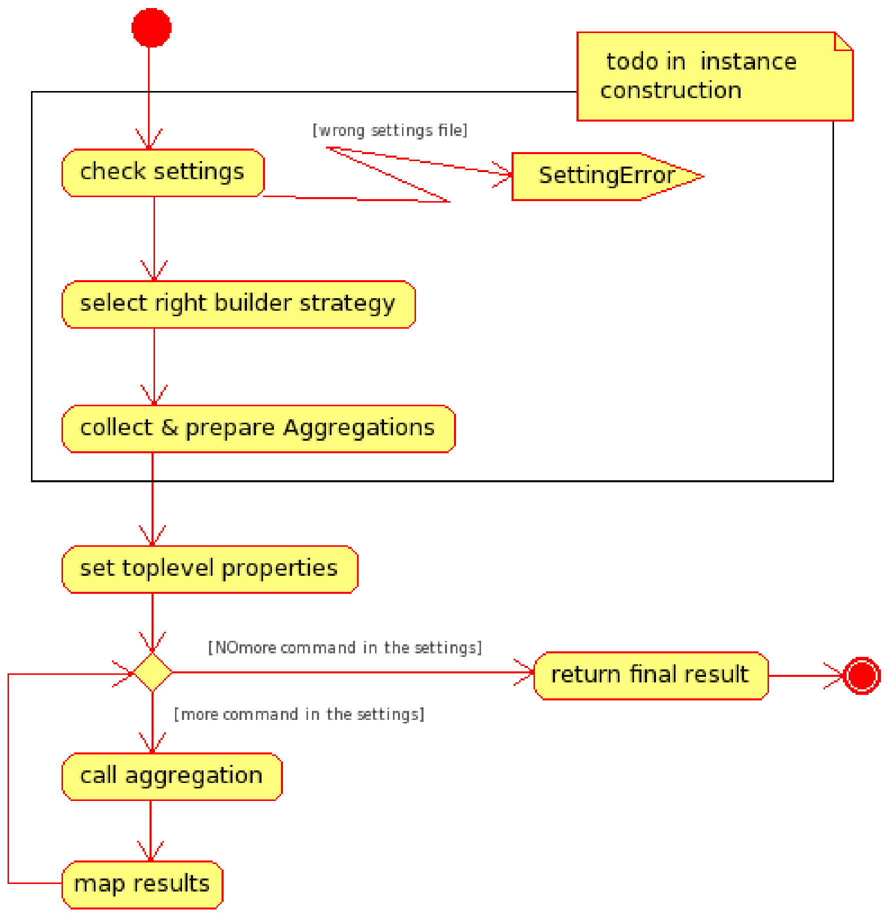

Figure 3.8: Actions involved in the *Building* execution step.

whole system. This is particularly relevant for the learner category, that can use this capability to better target the amount of information received. This approach allows the learner to fulfill its specific needs in term of richness of the personal feedback channel.

The active widgets are dynamically created at the top of the dashboard: every widget, based on a graphical template – mainly defined by the builder classes and specialised through the XML configuration – that defines its main aspects, represents an encoding of a single indicator (i.e. a piece information that can be relevant for the user activity) for each of the characteristics (one or more) of the learner profile. Each characteristic follows the presented path, from extraction to aggregation to being ready and usable for encoding into an indicator. This way, these widgets are the final outputs of the application.

3.1.7 Adaptivity

In the current implementation, the two upper layers (aggregator and builder) can be enhanced with adaptive features. As already mentioned, after the extractor layer has retrieved raw data from the sources the aggregator takes charge of merging and filtering the data in order to extract more refined information. This aggregation is based on the model that the instructional designer wants to provide to the learners and reflects the didactic approach adopted in the course.

With such an architecture, the support to the learning process based on adaptive profile externalisation can be achieved by adapting the visualisation to the specific didactic model.

The adaptivity is modeled in the configuration files for the two levels (aggregator and builder) through a simple XML. Its XML schema supports the conditional construct ”IF ... THEN ... ELSE ... ” which allows the GVIS visualisation to produce a different behavior with different properties. Furthermore, the schema allows each branch to be a leaf or another conditional, supporting in this way even a multilevel logic. The next example shows the usage of two levels. The properties can be any combination of source data values, on which a set of mathematical and logical operators can be applied. For instance, it can be decided that a particular widget may show a comparison of the level of knowledge of a student among the class only if his current knowledge level is greater than a threshold value; or we may want to show a particular widget only to the course instructor and not to the learners. This is implemented by including conditional instructions in the XML configuration files of the aggregator and the builder. The configuration files may contain variables, logical and arithmetical operators i.e. we have implemented the common comparison operators (more than, less than, equal and different) and the logical operators *AND*, *OR*, *XOR* (exclusive or), *NOT* (!).

The following example shows a possible condition:

```
<cond>
  <op>(v1 AND ((A &gt; 3) OR !(z)))</op>         // FIRST LEVEL CONDITION
  <operands>
    <val id="v1">CourseX.Concepts.list</val>
    <val id="z">CourseX.Student.count</val>
    <val id="A">CourseX.ConceptA.mean.knowledge</val>
  </operands>
  <true>...</true>
  <false>
    <op>(h &lt; t)</op>                          // SECOND LEVEL CONDITION
    <operands>
```

```
      <val id="t">CourseX.ConceptA.mean.knowledge</val>
      <val id="h">CourseX.ConceptA.userH.knowledge</val>
    </operands>
    <true>...</true>
    <false>...</false>
  </false>
</cond>
```

In the code above there are two conditions, and their interpretation is the following:
COND1 => *If the list of concepts in course X is not empty and either the average knowledge of concept A is greater than 3 or there are no students subscribed to the course, then display a particular widget, otherwise check another condition.*

The second condition is related to the knowledge level of the current learner, implemented in the *false* branch of the first one, and states the following:
COND2 => *If the knowledge level of the current learner is lower compared to the average level for course X, ...*

The construct shown allows for the creation of conditions of varying complexity and can therefore be used to specify a number of different behaviors at the level of granularity required.

In the conditional expression any variable that was associated in the extractor with an input of the user model can be used, as well as variables that represent user preferences and user device configurations, if available and retrieved by the extractor layer.

As a case study, the GVIS software was fed data from different LMSes used in a controlled experiment, to support a distance learning course.

The following examples show possible configurations enhanced through adaptive segments. The adaptive behavior can be achieved by the aggregator and the builder, and is driven by course data (i.e. data not directly related to a single user, such as the number of concepts in a course) and/or user data (i.e. data about the learner e.g. the activities performed, the knowledge acquired or preferences expressed). All this data is either collected by the extractor or it can be explicitly declared by the learner through his/her preferences and personal settings which are stored in the DB of the educational platform.

The first example, based on *user* data (see Fig. 3.9), adapts at the aggregation level. It presents the knowledge achieved by a student over the concepts of the course.

In Fig. 3.9 on the left is a compact view, where the average knowledge level of the learner (first column) over the concepts of the course is compared to the average knowledge level of the class (second column).

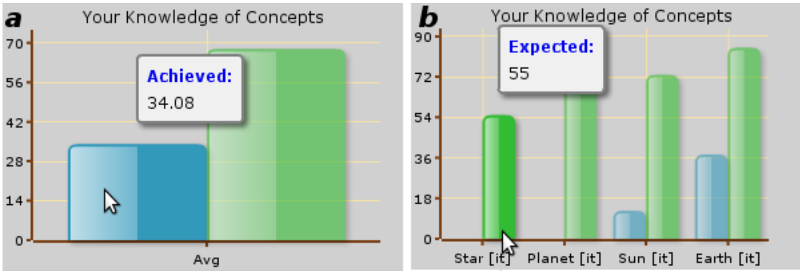

Figure 3.9: Example of the produced widgets: different aggregation level of the same data.

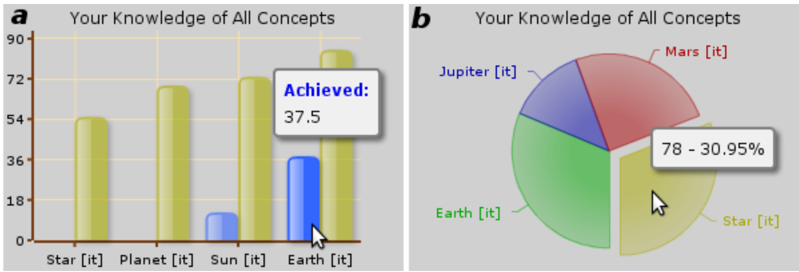

Figure 3.10: Example of the produced widgets: different graphical representation of the same base data, aggregated in different ways.

The right side of Fig. 3.9, on the other end, presents detailed information for each concept of the course (namely, *Star*, *Planet*, *Sun*, and *Earth*; each concept is followed by the *[it]* suffix to indicate that the content provided was adapted to the user's preferred language i.e. Italian). The choice around which of these two widgets should be presented to the user is made by the GVIS engine on the basis of the number of concepts to display: if the number of concepts is too large to be represented in the detailed visualisation, GVIS will present the aggregated view. It is important to notice that GVIS will only present data related to the concepts visited by the learner. This means that the adaptation rule can change the presented object automatically as the learner subsequently visits more concepts of the course. The threshold based on which one of the two alternative visualisations is chosen can be either a constant value in the XML configuration file or a value that is calculated on the basis of one or more characteristics of the class or activities done on the LMS, or even based on the device type currently used by the learner.

Another visualisation we developed, presented in Fig. 3.10, adapts the type of graph to the user preferences and to the number of concepts, in order to optimize the readability of the widget. It shows the information in two different formats: on the left hand side, a bar chart represents the knowledge levels of a learner over the concepts of the course, comparing

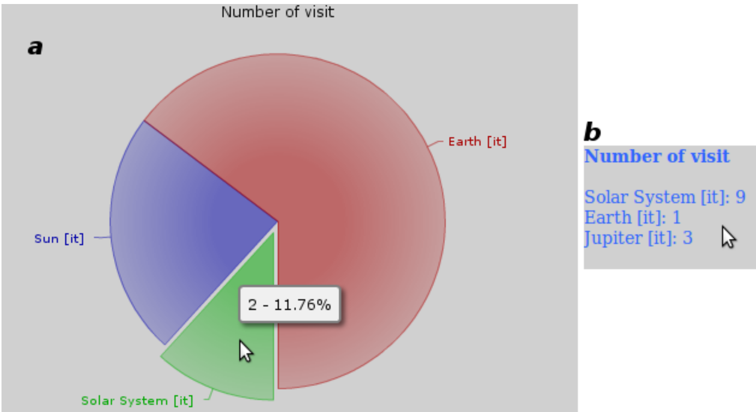

Figure 3.11: Example of the produced widgets: graphical versus textual representation.

these values with the expected knowledge level. On the right hand side, a pie chart offers a representation of the knowledge for every concept along with their relative weight in the total knowledge achieved on the course.

The next example, based on *course* data, is presented in Fig. 3.11 with an adaptation condition included at builder layer. This means that the data represented – like in the previous cases – is exactly the same, but here the encoding changes: in this case the adaptation is driven by either the type of hardware used by the learner or by the connection speed. A textual list is well suited for mobile phone devices or hand-held based platform, while a graphical widget (on the left) is more suitable for larger displays and broadband access.

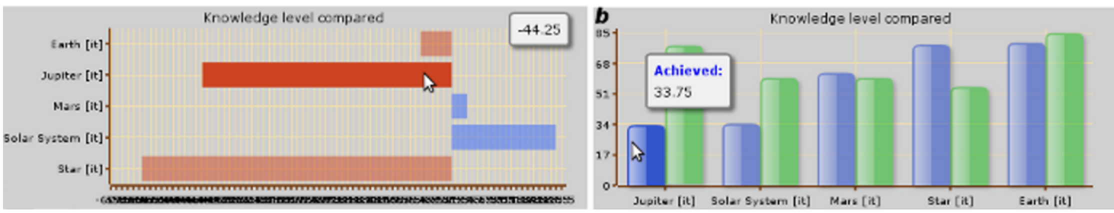

Figure 3.12: Examples of the produced widgets: differential versus absolute representation.

As previously stated, providing a way to open the profile to user inspection is important in the domain of Life Long Learning: the presentation of information about user activities and status as an indicator of the learning process is widely accepted as one of the key points to improve participation and increase the participants' satisfaction (Shneiderman and Plaisant 2004) and (Shneiderman and Plaisant 2005).

In some cases, the presentation of data is limited to a subset of the available information or

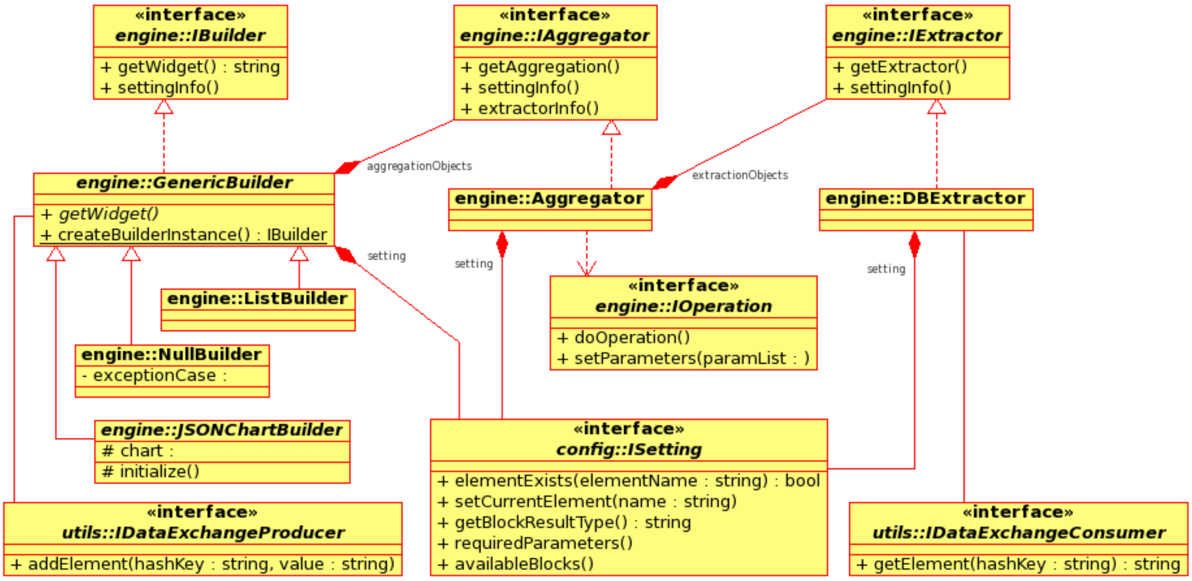

Figure 3.13: The class diagram for the *Engine* package (with some classes from other packages that have strong dependencies here).

is predefined by the developers and therefore fixed (Dimitrova 2003).

With GVIS, we aim to provide an easy way to create an effective graphical presentation of arbitrary subsets of data elements. The small examples of configurations presented and explained in the next sections are used to create widgets like the one shown in Fig. 3.12.

3.2 Development

In the following paragraphs the internal structure of the GVIS solution will be presented, describing the logic and presenting the communication mechanism with the environment implemented.

The main element is represented by the *Engine*, illustrated in Fig. 3.13, where the main class is represented by the *Engine::GenericBuilder* which implements the *Engine::IBuilder* interface. It relies strongly on the interfaces *Engine::IAggregator* and *Engine::ISettings*.

As it can be seen, the development process uses a design-pattern approach and enforces a clear separation between interfaces and implementing class, allowing for a clear distinction of the API[1] and for functional code. In Fig. 3.13 all the main components of the core – with

[1] *Application Protocol Interface* is the signature of the functions implemented and available use and represents the software interface.

the three level of Extraction, Aggregation and Building – are depicted together alongside with the relationship that exists between them: this clearly shows the interfaces they rely on for the mutual interconnection.

The top level, which is also the single point in charge of interacting with the learning environment for data exchange, is represented by the *Engine::GenericBuilder* and implements a static factory method, allowing for a static instantiation of the class without the need to create a full instance which would persist in the memory until it is explicitly destroyed. A generic builder exists in the *Engine::GenericBuilder* and every class that extends it should call this method to ensure the correct creation of all the contexts for the generation of a widget.

In adherence with this separation and encapsulation choice, all the classes implementing the *Engine::IOperation* interface rely on the same factory method to check the current environment (variables, configuration flags, parameters for the extraction and computation and so on).

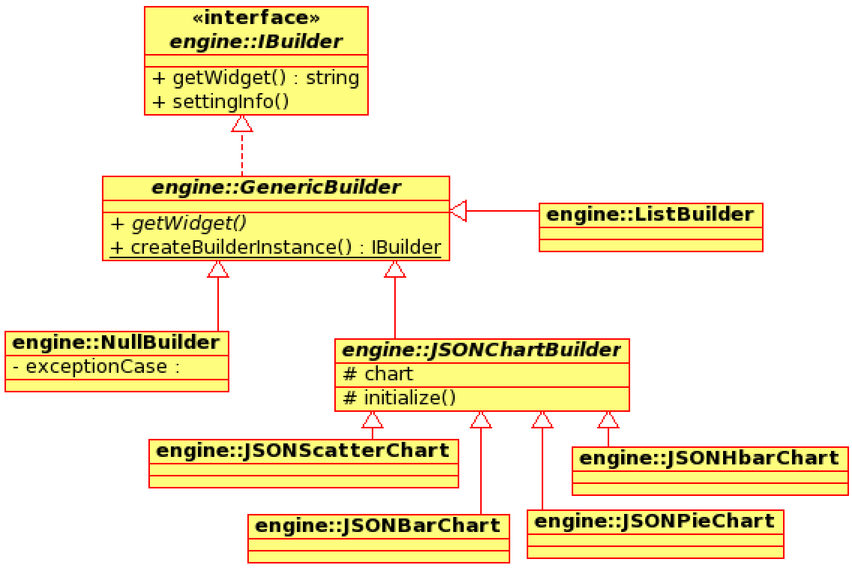

Figure 3.14: The class diagram for the *Engine::Builder* package.

In Fig. 3.14 a detailed view of the subpart of the *Engine* related to the *Builder* level is shown, highlighting how the *Engine::GenericBuilder* is specialised by different classes, each one of them being related to a specific type of widget (*Engine::NullBuilder* for the single textual values; *Engine::ListBuilder* for the textual lists, which requires some styling to be nicely presented inside an HTML context; and *Engine::JSONChartBuilder* that is the ancestor of all the classes used for feeding the chart based on a JSON representation of the data). Finally, the

Engine::JSONChartBuilder is further specialised by four classes: *Engine::JSONScatteredChart*, *Engine::JSONBarChart*, *Engine::JSONPieChart*, and *Engine::JSONHbarChart*. Each one of these classes format the data in a structure which is well suited to be used for a specific Flash[1] based chart implemented with Open-Flash-Chart library [2].

Operator	**Nr**	**Input**	**Output**
OpAverage	1	list	number
	2	list of lists	number
Calculate the average of the list of values given as input			
OpAvgComplex	-	list of lists	list of numbers
Calculate the average of the multi-lists of values given as input, generating a list of averages			
OpCount	1	list	number
	2	list of lists	number
Count the number of entries in the input data structure			
OpCountByField	1	list	number
	2	list of lists	number
Count the number of different fields present in the input parameter			
OpDiffer	-	2 lists	list
Calculate the difference of the lists received, as a set operator $\Rightarrow D = A - B$			
OpDistance	-	2 list	list of numbers
Calculate the distance of two lists in term of a list of single distance between corresponding terms			
OpExtractColumn	-	2 multidimensional list	list
Return a list made of a single dimension of the multidimensional received input			
OpExtractNameColumn	-	2 multidimensional list	string
Extracts the names of the dimensions from the array received in input			
OpExtractRow	-	2 multidimensional list	array
Extract a single record from the input multidimensional data structure			
OpFlat	-	2 multidimensional list	array
Transform the multidimensional input into an array			
OpFlatIn	-	2 multidimensional list	list
Transform the multidimensional input into a list			
OpLabelling	-	2 arrays	multidimensional array
Using the arrays received in input as values and names, crate a multidimensional array by merging them			

Table 3.1: The list of operators implemented in GVIS at the Aggregation level: part A.

Fig. 3.15 represents the level of *Aggregation* as another subpart of the *Engine*. The main

[1] FLASH is a technology originally developed from Adobe (R) to include active contents in an HTML context.

[2] Open-Flash-Chart is an open source library used to generate FLASH chart by simply providing the values and configurations in a object structure, encoded in JSON format – see `http://teethgrinder.co.uk/open-flash-chart/` .

Operator	Nr	Input	Output
OpIntersection	-	2 lists	list
Calculate the intersection of the lists received, as a set operator $\Rightarrow I = A \cap B$			
OpCompress	1	list	list
	2	list of lists	list of lists
Remove duplicates from the input list and return it			
OpLimitByNumber	-	2 list	list
Reduce the length of an input list to a certain number of elements			
OpLimitByValue	-	2 list	list
Reduce the length of an input list by eliminating the entries whose value is inferior to a threshold			
OpLimitByField	-	2 list	list
Reduce the length of a list by maintaining only the elements that match specific field names			
OpPassValue	-	2 list	element
Used to pass a value to the Builder			
OpRelativate	-	2 list	list
Transforms a list to make the values in it a percentage of the highest one			
OpRemoveCol	-	2 list	list
Returns the list received as input bar the dimension to be removed			
OpSimplify	-	2 list	list
Returns the list received as input bar empty or NULL values			
OpSum	1	2 list	number
	2	list of lists	number
Returns the sum of the elements in the input parameter			
OpSumAbs	1	2 list	list
	2	list of lists	number
Returns the sum of the absolute elements in the input parameter			
OpUnion	-	2 list	list
Calculate the union of the lists received, as a set operator $\Rightarrow D = A \cup B$			
OpEach	-	***	***
A special case operator that allows to iterate on Extractors based on data contained in a Aggregator			

Table 3.2: The list of operators implemented in GVIS at the Aggregation level: part B.

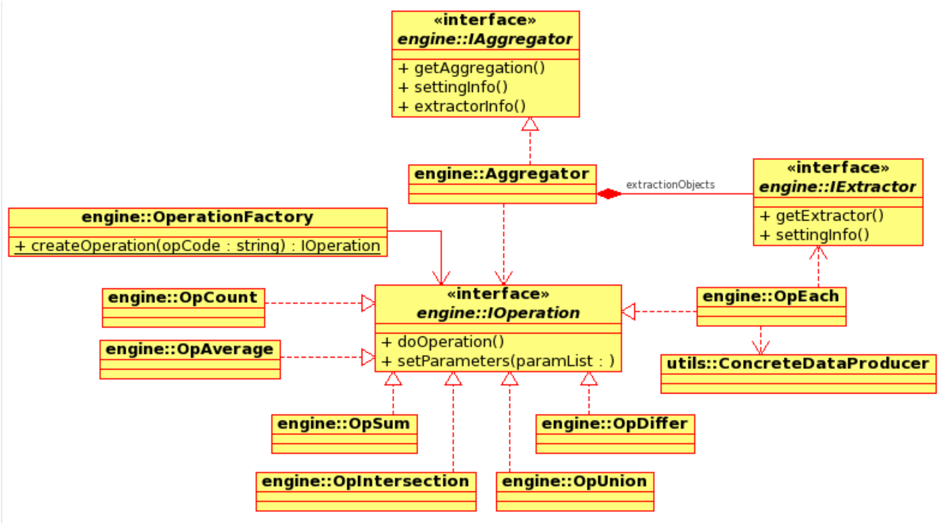

Figure 3.15: The class diagram for the *Engine::Aggregator* package.

class *Engine::Aggregator* is an almost empty implementation of the Interface *Engine::IAggregator* and has the only objective of including the correct implementation of the *Engine::IOperation* interface. This interface represents all the possible operations that the system can perform internally. The operations that exist are a basic set.

The operations can be categorised by the type of value returned – i.e. number, string or list – and by the number and type of input expected – i.e. order or unordered mono-dimensional list or multidimensional list – that cover all the basic needs for simple data manipulation.

In the category that takes a single list as input and a number as output the *OpCount* exists, which returns the number of elements contained in the list received as parameter; we also have the *OpAverage* that returns the numerical average of the list in input and the *OpSum* which behaves in the same way but returns the sum instead. For the type returning a result list we can enumerate the *OpInteraction*, the *OpDiffer* and the *OpUnion* which respectively return the common, uncommon and union of the record in the two unordered lists, that can be mono-dimensional or bi-dimensional (in the latter, the dimension on which to perform the operation should be declared in the operation configuration).

Tables 3.1 and 3.2 report the complete list of operators implemented, with an indication of the main data type expected as input as well as returned as output and a very brief explanation of the semantic of the operator.

Eventually, a special case (called *OpEach*) is implemented, which requires a list of values and an *Engine::IExtractor* instance as input. This operator performs the extraction of the second

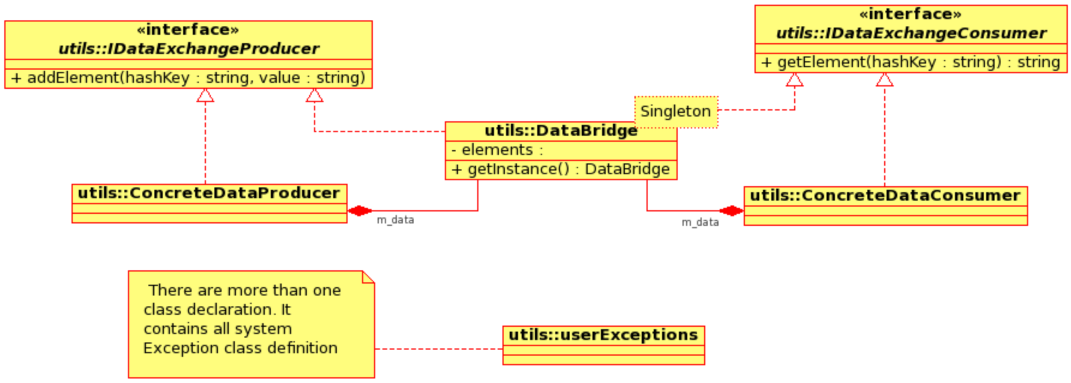

Figure 3.16: The class diagram for the *utils* package.

values using each one of the values included in the first input, resulting in either a single or multidimensional list based on the type returned by the *Engine::IExtractor* operation involved.

As it can be seen from Fig. 3.15 this last operation of the *Engine::Aggregator* relies directly on the interface for the extraction step and on a class on the *utils* package, called *utils::ConcreteDataProducer*, that supports the inversion of the normal data flow, which is usually strictly mono-directional from the extraction to the aggregation (see the next part of the *utils* package and Fig. 3.16 for details).

The package *utils*, whose main structure is depicted in Fig. 3.16, is a collection of helper classes that supports other packages. It provides common functionalities to all levels, such as the passage of the data between different levels and the management of unexpected status through the generation of exceptions.

For the sake of readability, in Fig. 3.16 only a generic *utils::userException* but not further specialisation class is depicted instead of the multiple and complex branch representing the specific exceptions returned by the package.

The most important class of the *utils* package is the *utils::DataBridge* singleton[1], which provides full support to the implemented *Engine* classes whether they need to exchange data between the levels of *Extraction*, *Aggregation*, and *Building*. In fact, it includes the two sides of the exchange, represented in the *utils::ConcreteDataProducer* (that implements the interface *utils::IDataExchangeProducer* and its public method *addElement* for providing new data) and in *utils::ConcreteDataConsumer* (implementing the interface *utils::IDataExchangeConsumer* and

[1]a *singleton* is a patter for OO-programming that restricts the number of instances of the object usable to a single entity.

its public method *getElement* for retrieving the data itself). It also allows for the correctness control of the data exchange on the sender's side as well as on the receiving ones, thus enabling a complete decoupling between levels and empowering the exchange mechanism by providing, if required, the possibility to also connect components that generate data to be used at a different level of the extraction.

The package *config* is accessible through the *config::ISetting* interface which is primarily implemented by three classes, each one of them dedicated to a specific level of the infrastructure as represented by the Fig. 3.17. A really useful helper class is the *config::ASetting*, which was exclusively designed for internal use – in fact, all of its methods are *private* – and provides the basic methods for parsing and binding the internal structure of an XML document.

The *config::ExtractorSettings* is the implementation of the lower level and provides a method *queryBlock* that, thanks to the inclusion of *config::-QueryBlockManipulation*, can provide a practical way to query a data source by reconnecting the formal query to the parameters used in the invocation phase. Moving on to the intermediate level of aggregation, the *config::AggregatorSettings* offers the functionalities required to support all of the operations on the parameters used, based on the inclusion of two classes i.e. *config::ExtractBlockManip* and *config::ComputeBlockManip*. These can handle the operation by either relying on the data coming from an extraction operation or the one generated from a different aggregation. By doing so they allow for the creation of a complex chain and enable the implementation of a sophisticated logic. At the top level, the *config::BuilderSettings* handles all the parameters (through the *config::ChartBlockManip*) and data configurations (including the *config::PropertiesBlockManip*)

When dealing with the XML settings, a special care has to be paid in order to avoid problems related to the interpretation of XML done by the support class developed in PHP[1]

Table 3.3 describes the implementation of the XML configuration that considered this issue by adopting an attribute, called *prevquery*, on the *param* tag. This attribute specialises the tag by indicating whether the parameter contained has to be interpreted as an actual one or if it is instead the index of the previous query to be used as input.

[1] *PHP* is one of the most widespread web programming language. The association of the formal and the actual parameter is done by positional substitution, which means that the first formal parameter is expanded in the first actual parameter before the related function is executed. This approach, despite its simplicity, can be prone to error if the association is not done respecting the order in a strict fashion. The interpretation of the XML segment is carried out using associative arrays (based on the name), but in the development two different tags[2] were mixed so as to generate two unrelated arrays. Reconstructing the original position of the actual parameters from this will require an additional meta-data structure, thus unnecessarily complicating the data structure and the configuration parsing operation.

Field	Value

XML source	`<parameters>`
	`<param>concepts.name</param>`
	`<param>knowledge.level</param>`
	`<param prevquery="true">0</param>`
	`<param>users.id</param>`
	`<param prevquery="true">10</param>`
	`</parameters>`

Interpretation	`[param] => Array`
	`(`
	`[0] => concepts.name`
	`[1] => knowledge.level`
	`[2] => 0`
	`[3] => users.id`
	`[4] => 10`
	`)`

Table 3.3: Positional interpretation of the XML parameters encoding.

In the following lines of code an example of the *Aggregation* and of the *Builder* are presented which show some of the peculiarities of the configuration:

```
<extraction>...</extraction>
<extraction>
%% REF TO THE FIRST VALUE TAKEN IN THE EXTRACTION LAYER %%
<toextract>Src1</toextract>
%% REF TO THE SECOND VALUE TAKEN IN THE EXTRACTION LAYER %%
%% IT IS DEFFERRED, BECAUSE IT DEPENDS ON OTHER EXTRACTOR RESULT%%
<toextract defer="true">Src2</toextract>
</extraction>
```

The extraction phase takes into account two sources from the Extractor (namely *Src1* and *Src2*): the first one is immediate, whereas the second one is to be delayed to use the parameters coming from an Aggregation phase (as indicated by the attribute *defer="true"*).

```
<computation>
  %% FIRST COMPUTATION %%
```

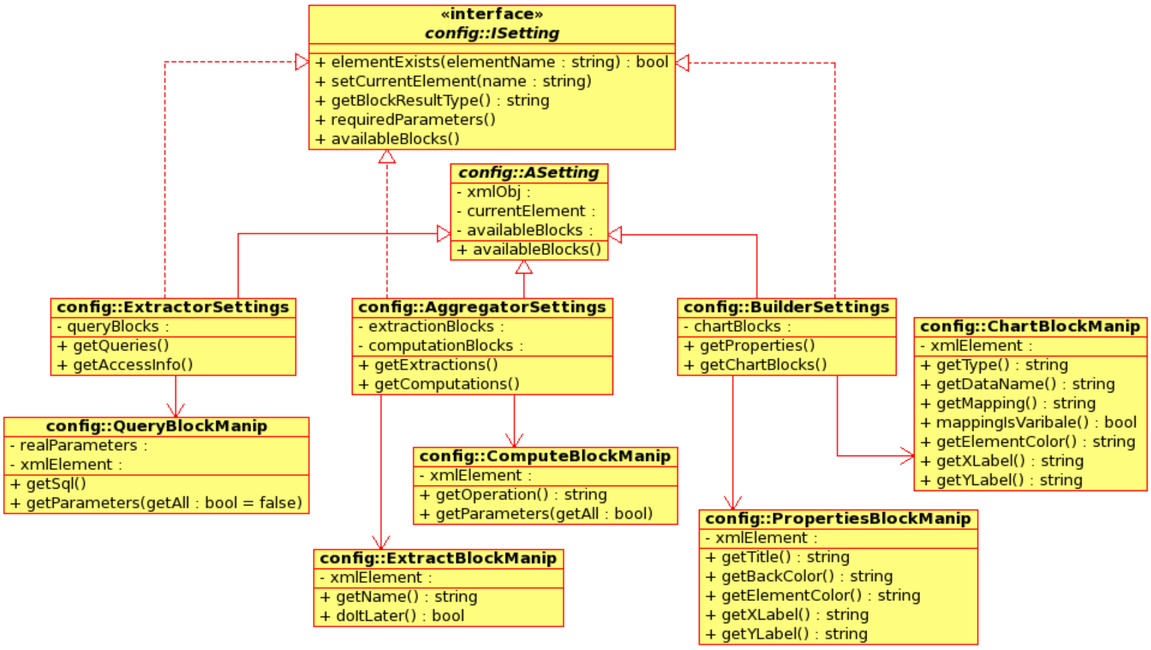

Figure 3.17: The class diagram for the *config* package.

```
<tocompute>
  <operation>intersection</operation>
  <parameters>
    %% FIRST VALUE TAKEN FROM THE EXTRACTION LAYER %%
    <param>0</param>
    %% FIXED VALUE, AS SPECIFIED %%
    <param fixed='1'>1</param>
  </parameters>
  %% TYPE OF RESULT RETURNED %%
  <resulttype>list</resulttype>
</tocompute>
%% END OF FIRST COMPUTATION %%

%% SECOND COMPUTATION %%
<tocompute>
  <operation>count</operation>
  <parameters>
    %% RESULT COMING FROM FIRST COMPUTATION %%
    <param computed="true">0</param>
  </parameters>
  %% TYPE OF RESULT RETURNED %%
```

```
    <resulttype>numeric</resulttype>
  </tocompute>
  %% END OF SECOND COMPUTATION %%

  %% THIRD COMPUTATION %%
  <tocompute>
    <operation>each</operation>
    <parameters>
      %%RESULT COMING FROM SECOND COMPUTATION %%
      <param computed="true">1</param>
      %% SECOND VALUE TAKEN FROM THE EXTRACTION LAYER, DEFERRED %%
      <param>1</param>
    </parameters>
    %% TYPE OF RESULT RETURNED %%
    <resulttype>list</resulttype>
  </tocompute>
  %% END OF THIRD COMPUTATION %%
</computation>
```

The computation is composed of three steps, the first one –using the first data source and a fixed value and identified by the relative attribute– gives a return type of list through an intersection. The second step consists of performing a count of the elements included in the list returned from the previous aggregation step. In the final computation, the XML instructs the infrastructure to use the results of the second step as an input parameter (one for each call) to instantiate the extraction known as *Src2*.

```
<properties>...</properties>

%%GRAPH TYPE %%
<chart type="bar">

  %% FIRST SOURCE TO BE USED FOR THE CURRENT GRAPH %%
  <chartsource>
    %%ELEMENT OF COMPUTATION TO BE USED %%
    <data from="system">StudConCount</data>
    %% MAPPING ON THE WIDGET %%
    <mapping>values</mapping>
    %% LABEL TO USE %%
    <label>Selected Student</label>
  </chartsource>
```

```
  %% OTHER SOURCE TO BE USED FOR THE CURRENT GRAPH %%
  <chartsource>
    <data from="system">ClassConCount</data>
    <mapping>comparison</mapping>
    <label>Class</label>
    <elementcolor>\#220505</elementcolor>
    ...
    </chartsource>
</chart>
```

The Builder relies on two different results returned from the Aggregation (namely *StudConCount* and *ClassConCount*) to generate two series, called "Selected Student" and "Class" respectively. The first one acts as the main value series whereas the second ones acts as the comparison series. The second series is filled with the color #220505 and the chart type requested is a vertical bar chart.

Fig. 3.18 and Fig. 3.19 represent the interaction diagrams for the widget generation inside the environment as seen by the dashboard and a final user of the GVIS tool respectively.

An example of the JSON result[1] communicated to the dashboard for the widget instantiation is presented in the next segment, where a bar chart is requested and returns six results.


```
{
  "elements": [
    {
      "type": "bar",
      "values": [
        1,
        2,
        3,
        4,
        5,
        6
      ]
    }
  ],
  "title": {
    "text": "Personal visited concepts count"
  }
}
```


[1] for the full syntax of the JSON answer see Appendix B.

Finally the appearance of the dashboard implemented in the original GVIS infrastructure is depicted in Fig. 3.21. In the next chapters, different usages of the GVIS infrastructure will be presented and the capabilities and flexibility achieved in the developed code will be shown.

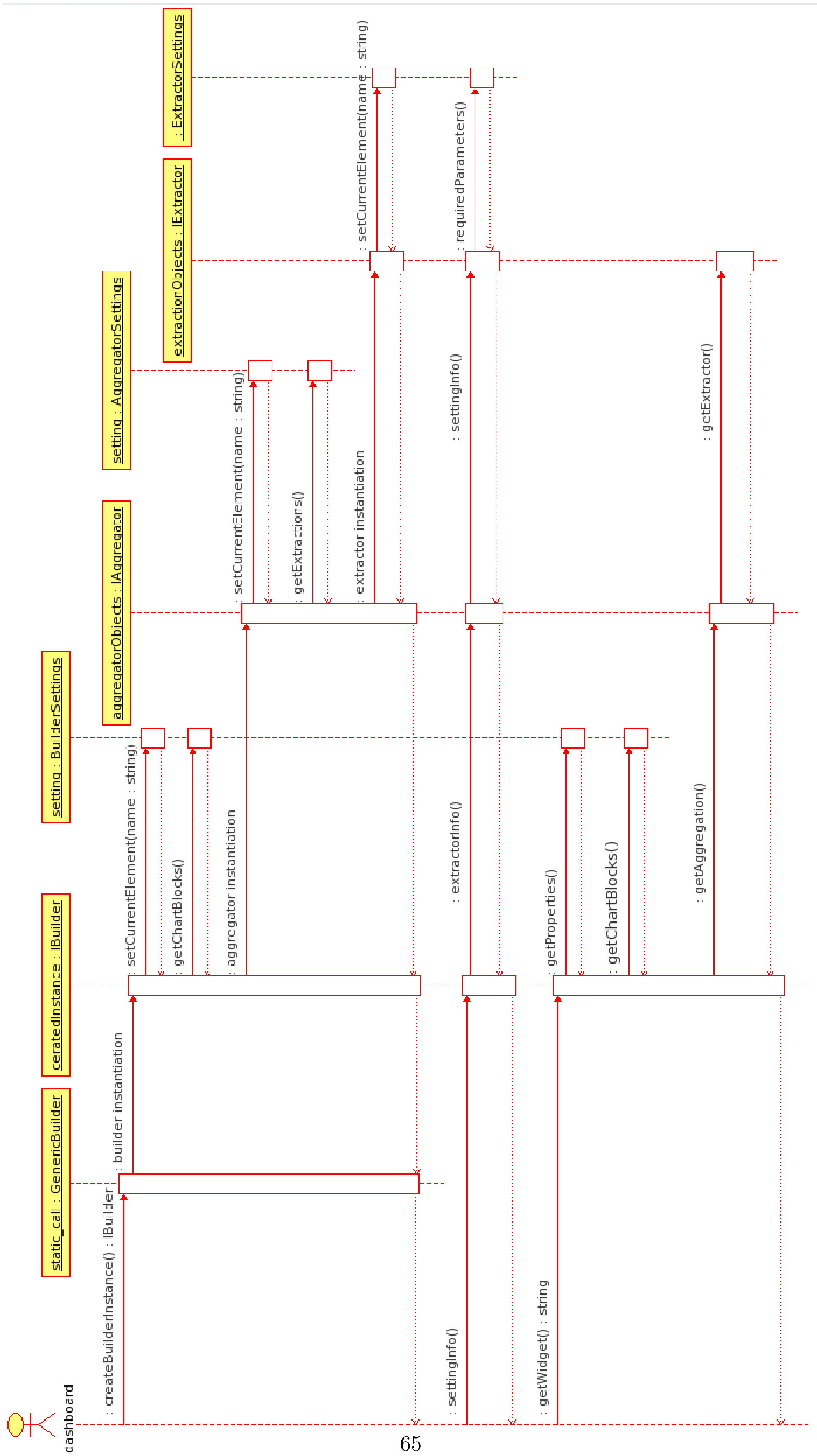

Figure 3.18: The sequence diagram for the widget creation (as seen internally: Dashboard).

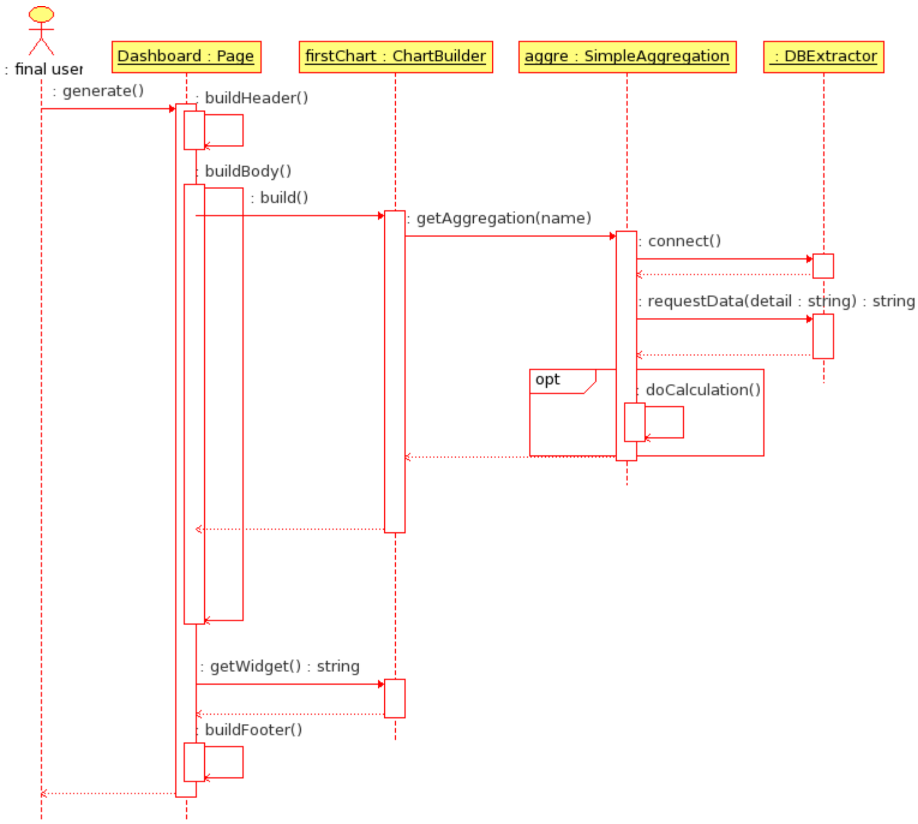

Figure 3.19: The sequence diagram for the complete widget creation (as seen externally: user).

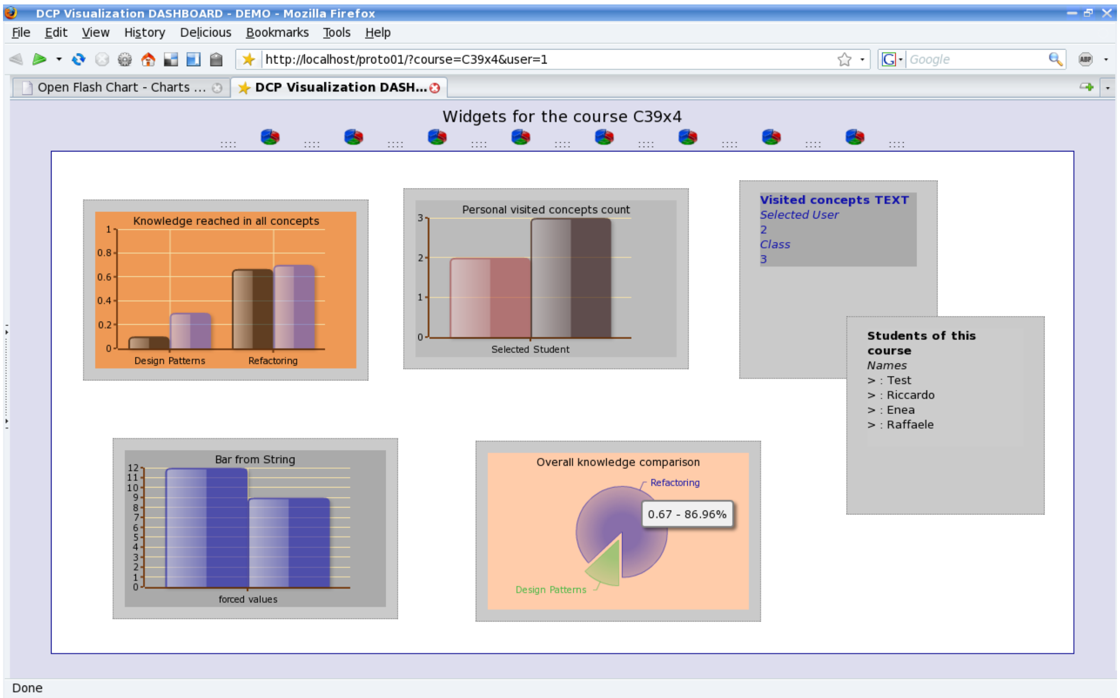

Figure 3.20: The dashboard as seen in the original conception of the *GVIS* infrastructure.

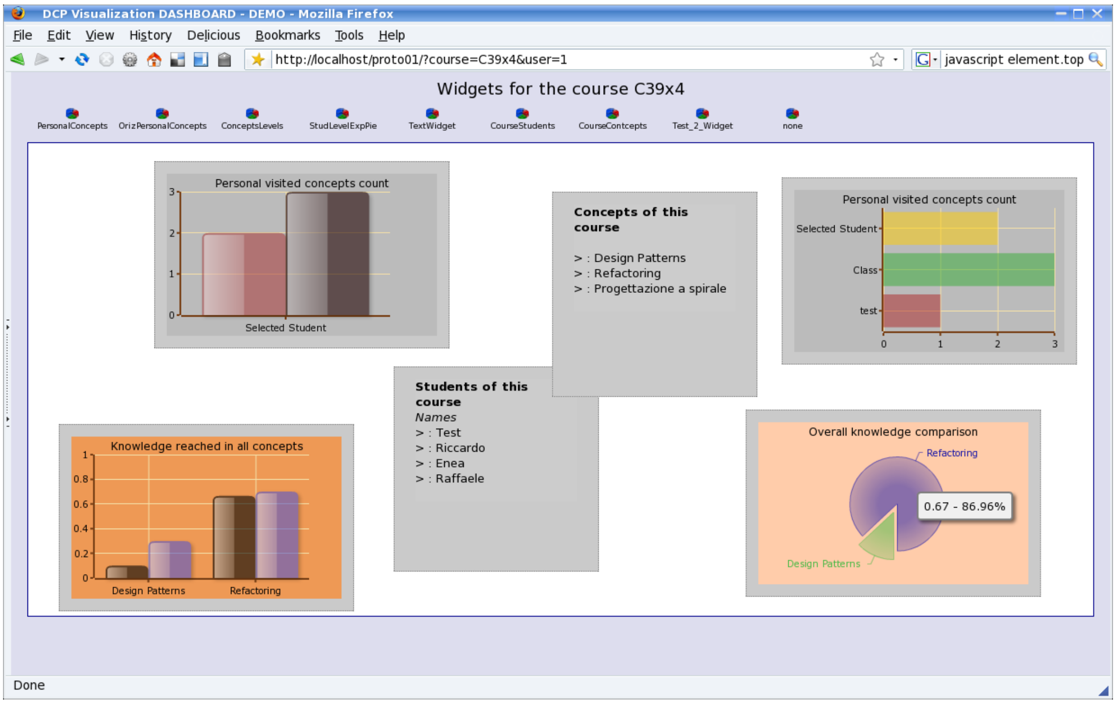

Figure 3.21: Another view of the original conception of the *GVIS* infrastructure.

Chapter 4

GVIS: evaluation in GRAPPLE

This chapter presents the evaluations produced based on the experiments that were carried out on the design and implementation of the infrastructure. The first evaluation was performed using mock-ups, while practical experiments followed in the next evaluation. The evaluations performed in this chapter aimed at support the design and initial implementation of the visualizations to be provided as the interface of GVIS with the users. On the other end, the feasibility and integration with other LMS and other heterogeneous data sources will be presented in next chapter.

4.1 GVIS in the context of GRAPPLE

As already stated, GVIS was developed to become one of the multiple services in the GRAPPLE infrastructure, and in this precise context the user requirements were gathered, the design was conceived and a first evaluation was produced. The first step was based on mock-ups, in order to receive the users comments and feedbacks about the perceived usefulness, the potential issues, but also suggestions for improving our design.

4.1.1 GRAPPLE User requirements

As part of the initial activities on the GRAPPLE project a number of interviews and meetings have been conducted on requirement gathering and analysis [1]. The questions specifically conceived for the evaluation of the GVIS module were integrated in a questionnaire with a broader

[1] The results of the interviews conducted with a series of end users involved in the academic and industrial settings can be found in the GRAPPLE project public deliverables D10.5 (Pekczynski, Steiner, Mazzetti, and Barak 2011) and D9.1 (Harrigan, Kravcik, Steiner, and Wade 2009).

objective, including also aspects of adaptivity. For these reasons only a part of the questions was usable to run an analysis of preferences and needs with respect to visualisation. Nevertheless, some of the results obtained are still relevant to the present work. From the analysis of the open answers and comments documented in the interview summaries, it can be seen that the visualisation of acquired knowledge and learning progress is considered to be important (Harrigan, Kravcik, Steiner, and Wade 2009). Many considered the visualisation of the various stages of learning and how they are adapted an interesting prospect. One learner raised the question as to how he –or she– can know "if they haven't missed anything important". Learners need a visual map of where they are and where they are going with each important milestone clearly marked. The training providers and tutors also require a visualisation of the learners progress and of the structure of the course as it currently stands.

Another characteristic emerged as important is the expected "transparency" of the user model from the learners side, that should be more similar to a personal notebook than to a black box used only by one or more adaptive learning systems. This fact could also help in seeing it under a more positive light, as a learners portfolio. For reaching this objective, the learner should always be able to inspect its contents, at least to some degree (Pekczynski, Steiner, Mazzetti, and Barak 2011). The system should be able to point out gaps between what a learner thinks that he/she knows (perhaps through voluntary questionnaires) and what he/she actually knows as portrayed by tests and exercises, which can help the learner to identify its own strengths and weaknesses. The system can also point out the discrepancies (if any) between an optimal learning path (i.e. the one designed by the instructor) and a preferred learning path as indicated by the learner (through the followed contents). To a lesser degree, it may also be used to match the learners with their peers who are strong in areas they are weak in, and vice versa (Harrigan, Kravcik, Steiner, and Wade 2009).

These findings are in line with results obtained from responses of stakeholders on the adaptation criteria, which showed that the user characteristics deemed most important for adaptation are **learning goals**, **tasks**, and **learner knowledge**. These are actually the main user model variables in direct relation to the learning tasks. In addition,an appropriate visualisation can be assumed to be effectively supportive for the user's learning process in an adaptive system. In order for the learning process to be successfully tailored to the variables above, having appropriate visualisations of the respective information in addition to the adaptation process itself is certainly a plus. This shouldn't come as a surprise as it already emerged from the initial requirements gathering activity conducted with the users. From the above results and considerations as derived from the requirements analysis, the following requirements can be derived: The visualisation has to make clear what the **main parts and concepts** around the course are. Important milestones, in terms of **learning objectives**, should be also clearly represented. Learners should be able to **inspect the content of the user model** and see what their current status is, as well as how they are **progressing** with respect to the learning objectives defined by the teacher. Any **discrepancies** between the learning objectives and the current status of knowledge of the learner as inferred by the learner model should be clearly reported by the visualisation. This could help the learner **identify his/her strengths and weaknesses** and **promote meta-cognition**. Tutors and instructors need to access this information to **monitor the learners progress** and review the **structure of the course** as it currently stands. **Peer learner models** should be provided. This will help learners **find "complementary" peers** who are strong in areas in which they are weak, and vice-versa.

4.1.1.1 Information descriptors

This section contains details about the information that GVIS has to provide to both learners and instructors. The following information has been compiled taking into consideration the source data identified in the previous section and the user requirements and analysis defined in the deliverable D4.5a (Mazza, Mazzola, Glahn, Verpoorten, Nussbaumer, Steiner, and Heckmann 2009). The overall goal of the following information is to support the learners to acquire meta-cognitive skills and the instructors to monitor the status of their students. For each information item the following is indicated:

- A short name for the data (column *Name*) and a code to describe whether the information represents facts about the learner or about the class (column *Type*).
  - L: about learner and peers
  - S: about the class
  - X: about both
- The name and a synopsis of the meaning of the data (column *Information*)
- Target users (column *Main target users*)
- Description of the required source data (column *Required source data*)

- Possible interaction of the users with the visualisations that represent that information (column *User interactions*).

Two or more pieces of information described in the following tables (Table 4.5 and Table 4.6) can be combined into a single graphical representation. Moreover, we used the previously presented coding schema to describe whether the information represents facts about the learner or about the class.

Two types of visualisations are generated by GVIS: a compact indicator, that can be placed as side block on the main page of the course, and a detailed visualisation, placed on a special dashboard that can be accessed by the system users by clicking on the compact visualisation.

4.1.1.1.1 Descriptors for compact indicators The information described in the following table will be implemented as compact indicators that the users will see as side block on the main page of the course in the Learning Management System. A compact indicator can implement one or more of the items described in the Table 4.1.

4.1.1.1.2 Descriptors for Dashboards views The information described in the Table 4.2 and Table 4.3 is implemented with a special dashboard that can be accessed by users of the system by clicking on one of the compact visualisations described in the previous section. The dashboard contains the widgets that implement the items described in already mentioned tables.

4.1.1.2 Data input

The design of visualisations requires a well-defined information around the type of data to represent. This is important for three main reasons: the data type determines the visualisation to be used, the data type determines what data sets can get combined in one visualisation, and finally the data type partially reflect the meaning of the information provided.

The following input data types were identified:

1. **Text** - textual information represented as a sequence of characters
2. **Number** - numerical information, which can be of type Integer or Float
3. **List** - ordered sequence of data of type Text or Number. List elements are identified as Integers. It can also be a list of records, where a record is a sequence of Texts or Numbers.

Name	Information	Main target users (in order of relevance)	Required source data	Description of the aggregations
ConVisStud (L)	Number of concepts visited by a learner	Learner	ConList, ConVis	Counts the number of concepts of the course (ConList) visited one or more times by the learner (ConVis)
ConVisClass (S)	Number of concepts visited by the class	Instructor, Learner	StuList, ConList, ConVis	The average value of concept visited (ConList, ConVis) per learner (StuList)
KnowStud (L)	Overall knowledge acquired by a learner	Learner	ConList, KnoLev, KnoRange	The average level of knowledge level (KnoLev) of the learner per concept of the course (ConList)
KnowClass (S)	Overall knowledge acquired by the class	Instructor, Learner	StuList, ConList, KnoLev, KnoRange	The average of the overall knowledge level (KnoLev, ConList) per student of the course (StuList)
ActStud (L)	Number of activities performed by a learner	Learner	ActList, ActVis	Counts the number of activities of the course (ActList) performed one or more times by the learner (ActVis)
ActClass (S)	Number of activities performed by the class	Instructor, Learner	StuList, ActList, ActVis	The average of the number of activities performed (ActList, ActVis) per learner (StuList)

Table 4.1: Descriptors for GRAPPLE compact indicators

Name	**Information**	**Main target users (in order of relevance)**	**Required source data**	**Description of the aggregations**	**User interactions**
GoalStudNum (X)	Number of learners that have achieved a particular learning goal	Instructor	StuList, GoalRea	Counts the number of learners (StuList) that have achieved a learning goal (GoalRea)	
KnowConcStud (L)	Knowledge level of the learner for every concept of the course	Learner, Instructor, Tutor	ConList, KnowLev, KnoRange	Given a learner, it provides, for each concept of the course (ConList), the current level of knowledge acquired by the learner (KnoLev)	Filtering on concepts. Sorting by knowledge
KnowConcClass (S)	Knowledge level of the class for every concept of the course	Instructor, Tutor, Learner	StuList, ConList, KnoLev, KnoRange	Provides the average knowledge level (KnoLev) per student (StuList) for any concept of the course (ConList)	Filtering on learners, goals. Sorting by knowledge
KnowConcExp (X)	Expected knowledge level for every concept of the course	Learner, Tutor, Instructor	ConList, KnoLev-Exp	Provides information about the expected knowledge level (KnoLevExp) for every concept of the course (ConList)	Filtering on concepts
KnowConc-StudHist (L)	Knowledge level of the learner in the past for every concept of the course	Learner, Tutor, Instructor	ConList, KnoHist, KnoRange	Provides, for each concept of the domain (ConList), the level of knowledge acquired by the learner at a particular time in the past (KnoHist)	Manipulation of date (eg. through a slider)

Table 4.2: Descriptors for GRAPPLE dashboard's views - part 1

Name	**Information**	**Main target users (in order of relevance)**	**Required source data**	**Description of the aggregations**	**User interactions**
KnowConc-ClassHist (S)	Knowledge level of the class in the past for every concept of the course	Instructor, Tutor, Learner	StuList, ConList, KnoHist, KnoRange	Provides, for each concept of the domain (ConList), the average level of knowledge per student (StuList) at a particular time in the past (KnoHist)	Manipulation of date (eg. through a slider)
SuitConcStud (X)	Indication about which concepts are currently suitable for the learner	Learner, Tutor	ConList, ConSuit	Indicates which concepts (ConList) are currently suitable (ConSuit) for the learner	Filtering on concepts
SuitActStud (X)	Indication of which activities are currently suitable for the learner	Learner, Tutor	ActList, ActSuit	Indicates which activities (ActList) are currently suitable (ActSuit) for the learner	Filtering on activity (activity type?)
ActVisStud (L)	Indicates which suitable activities have been visited by a student		ActList, ActVis	Indicates which activities (ActList) have been visited (ActVis) by the learner	Filtering on activity (activity type?)
GoalStud (L)	Indication of achievement for each learning goal	Learner, Tutor, Instructor	GoalList, GoalRea	Indicates which learning goals (GoalList) have currently being achieved (GoalRea) by the learner	Filtering on goal

Table 4.3: Descriptors for GRAPPLE dashboard's views - part 2

4.1.1.2.1 Expected input data The data providers are the components in the GRAPPLE infrastructure where the input data for GVIS is located. We identified two main data providers: GAT and GUMF. Table 4.4 presents them alongside with a description and connection interface to the GVIS.

Data provider	**Description**	**Interface to GVIS**
GAT	The GAT (GRAPPLE Authoring Tool-set) is the authoring component of a GRAPPLE course. It is composed by three elements i.e. DM (Domain Model), CRT (Concept Relationship Type) and CAM (Conceptual Adaptation Model). Some of the data from these elements, defined by the authors/instructors by means of these components, are needed by the visualisations (such as the list of concepts for a course or the expected knowledge level for specific concepts). The GAT has an internal repository that contains all the data structures. Data from this repository can be extracted by means of specific web-service interface.	GVIS fetches GAT data through a web-service interface provided by the CAM component
GUMF	Contains all the user-related data. It is the main source of data for GVIS. In particular, GUMF contains the data generated by GALE, such as whether a concept is currently suitable for a user or not, the level of knowledge of learner over the concepts, and by LMSes, such as the role of the user in a course, the list of students, the list of activities. GALE and LMS store their user data into GUMF through the Grapple Event Bus (GEB).	GVIS connects to the GUMF GRAPPLE Event Bus.

Table 4.4: Data providers for GVIS

GVIS is connected to GAT and GUMF by mean of a specialised interface and a general Event Bus. Every other data provider in GRAPPLE will pass through the GRAPPLE Event Bus (GEB) and will be stored in the GUMF facility, where GVIS can access it.

In the next paragraphs, the types of information expected are described.

Table 4.5 and Table 4.6 show the data needed by the visualisations. All this data is provided to GVIS by one of the data providers identified in Table 4.4.

4.1.2 Evaluation on mock-ups

Before starting with the implementation of visualisations, some of the mock-ups of the visualisations that we have planned to implement for GVIS were conceived. The aim of these mock-ups

Name	Description	Data Type	Data source	Comments
ConList	List of all the concepts of a course	List	GAT (CAM)	
ConSuit	Whether a particular concept is suitable for a user or notation	Boolean	GUMF (GALE)	This information is inferred by GALE, and then stored into GUMF
ConRel	List of existing domain relationships between two concepts	List	GAT	For future use. This source data is currently not used in information descriptors
StuList	List of the student enrolled on a course	List	GUMF (LMS)	
StuAccess	The number of times a user has accessed a course	Integer	GUMF (LMS)	For future use. This data source is currently not used in information descriptors
ConVis	The number of times that a user has visited a concept in GALE	Integer	GUMF (GALE)	
UsrRole	Provides information about the role of a specific user on the course (e.g. teacher, learner, tutor, etc)	Text	GUMF (LMS)	
KnoLevel	Knowledge level of the student on a particular concept	Number	GUMF (GALE)	
KnoHist	Knowledge level of the student on a particular concept at a particular point in time	Number	GUMF	
KnoLevExp	The expected knowledge level of the student on a particular concept	Number	GAT (CAM)	The expected knowledge level is defined by the instructor and corresponds to the minimum knowledge level that the learner has to acquire in order to have a sufficient knowledge of a concept to finish a course. A default value is be assumed and the author will replace the default values with CRTs if he/she wishes to do so

Table 4.5: List of source data for GVIS - part 1

Name	Description	Data Type	Data source	Comments
KnoRange	Range of values (minimum and maximum values) for the knowledge level	Numbers	GUMF	We expect that the range of values for the knowledge level to be [0.0, 1.0]. However, if the range is different, the GUMF should provide us with the adjusted values
GoalList	List of all the goals defined for a course	List	GAT (CAM)	A goal represents a particular state reached by the learner during the learning process. For instance, it could indicate that a learner reached a milestone, or acquired enough knowledge of the course topics. Each goal corresponds to a particular CRT. There can be several goals. Goals are set by the course authors in the CAM editor. This goal list is by default the first thing presented to the student when entering the course. If he/she does not want to have it displayed anymore, they can tick the appropriate option. The default implicit goal is: reach the expected knowledge levels for all concepts. Additional goals can be set via CRTs
GoalRea	Whether a user reached a particular goal	Boolean	GUMF (GALE)	A goal is defined using a particular User Model variable. The variable is defined in the CAM and calculated by GALE
ActList	List of the learning activities on a course. A learning activity is an action in a system or an environment that is related to the learning	List	GUMF (LMS)	Currently, only LMSes actions are tracked into the GUMF
ActVis	Number of times that a user visited a particular activity (from ActList)	Integer	GUMF (LMS)	

Table 4.6: List of source data for GVIS - part 2

was to evaluate them with the target user groups. Once the evaluation was concluded, the widgets were implemented, taking into account the results of the evaluation run, but also the most relevant feedbacks received by the users. This process from the conception to the evaluation of the visualisations will be presented in details in this section.

The original GVIS module supports two kinds of widgets: compact indicators, which are small widgets integrated into the LMS course/ITS web interface, and widgets, to be placed into a dashboard, that open a new window in the user interface.

The visualisations can be divided into 3 groups, according to the type of information they provide:

- Knowledge
- Activities
- Goals

In the next paragraphs, after the indication of the expected input data, the conceived mock-ups will be presented in details, together with their explanations. Later on in the chapter, an analysis of the collected feedback is outlined.

4.1.2.1 Knowledge

The knowledge level around concepts of the course is the most important information that has to be presented to the users by GVIS. It can be presented to the learner but also to the instructors and tutors. The overall knowledge acquired by a learner could be represented by the average (either weighted or not) of his/her knowledge level on each concept of the course. This means that the representation can be a simple average or it can also take into account the importance of, or the time spent on each subject, to create the weighted ones.

4.1.2.1.1 Views The first type of view is the overall knowledge level. The information represented by the following views is referred by the information descriptors as:

1. *KnowStud*: The overall knowledge acquired by a learner
2. *KnowClass*: The overall knowledge acquired by the class

The pictures presented in Figg. 4.1 and 4.2 encode this information in two forms i.e. a textual and a graphical form (both available for integration in the education system interface

Figure 4.1: The overall knowledge level in textual view.

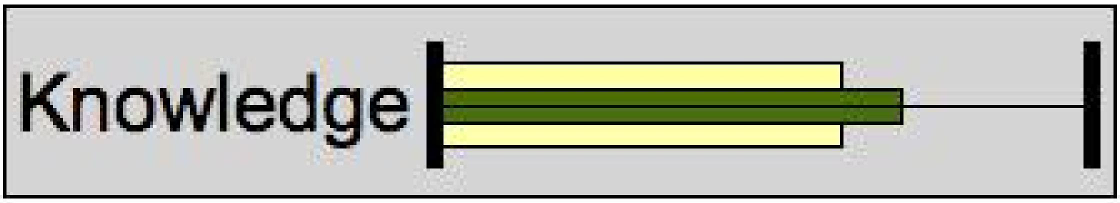

Figure 4.2: The overall knowledge level in graphical view (learners' version).

where the learning experience is provided with). In order to provide an integration into the LMS interface (usually web-based applications), the HTML document object model is used. In fact, the GVIS output uses an HTML segment that works as a proxy, with a twofold objective. First of all, it is able to create the bridge needed for accessing, in the LMS environment, the current user and the course data; at the same time, it is capable of allocating the necessary physical space to contain the visualization.

In the graphical view depicted in Fig. 4.2 we also represent (in yellow) the overall knowledge acquired by the class, where the green bar represents the learner's knowledge. On the instructor/tutor view, only the yellow bar is represented. The end point of the line represents the upper bound of the range of the knowledge level, as defined by the Instructional Designer.

When a user clicks on the widgets a more detailed view is provided. This is primarily important because more detailed views are in charge of representing the knowledge level for every concept of a course, allowing autonomous user exploration of this informative space. To this end, a dashboard is provided to users of the GRAPPLE system that contains more detailed visualisations and can work as a container for all the widgets provided, regardless of whether they are automatically represented or called on user request. This solution also allows the user to resize and reorder the informative widgets in the dashboard area.

In particular, the detailed views provide a way to represent the following information descriptor:

- *KnowConcStud*: Knowledge level of the learner for every concept of the course.

- *KnowConcClass*: Knowledge level of the class for every concept of the course.
- *KnowConcExp*: Expected knowledge level (target) for every concept of the course.
- *KnowStud*: The overall knowledge acquired by a learner (as an average of the ones acquired on each concept of the course by that learner).

The first visualisation we designed is composed of two overlaying bar charts (see Fig. 4.3), where each vertical bar represents a concept of the domain. The knowledge acquired by the student is encoded into the green bar while the knowledge acquired by the class is represented by the yellow bar. If the expected knowledge – i.e. the level that the ID requires for the learner to achieve for a specific concept covered by the course – is available, then it is represented with a horizontal, black bold line. The dashed line represents the maximum knowledge level achievable. The level is expressed as a percentage, if not otherwise stated in the data provided: this is useful to rescale the bars and gives a global and comparable view on all the concepts. This view was designed for being available to learners, tutors and instructors.

If historical data is available in one or more data source (in the GRAPPLE case, it could be the GUMF component), it could be interesting to compare the status of the past knowledge of a student with his/her current knowledge. The next visualisation (Figure 4.4) was intended to compare the knowledge level of a student and the class at a particular time in the past (bar graph series above) with the current status (bar graph series below). This visualization was mainly intended for learners, but also instructors can take advantage from such a time comparison through an overview of the single learner/class temporal evolution in terms of knowledge. Also, the expected knowledge level can vary in time, based on the didactic approach and the adjustments done by the ID.

A different view was specifically conceived to be provided to tutors, in order to allow a more specific comparison of concepts and learners. The visualization is based on a matrix, where the learners are mapped into columns and the concepts into rows (see Fig. 4.5). The level of knowledge of a student in a concept is mapped into the dimension (in particular, the radius) of each cell of the matrix. The color of the cell represents whether the student reached (in green) or didn't reach (in red) the target knowledge level.

In another visualisation (see Fig. 4.6), the bar graph near the concept names gives an indication of the overall knowledge of the class on that concept (red bar). The shape of the cell represents whether the student reached (square) or didn't reach (circle) the target knowledge level. The green bar next to the learner names indicates the overall knowledge acquired by the

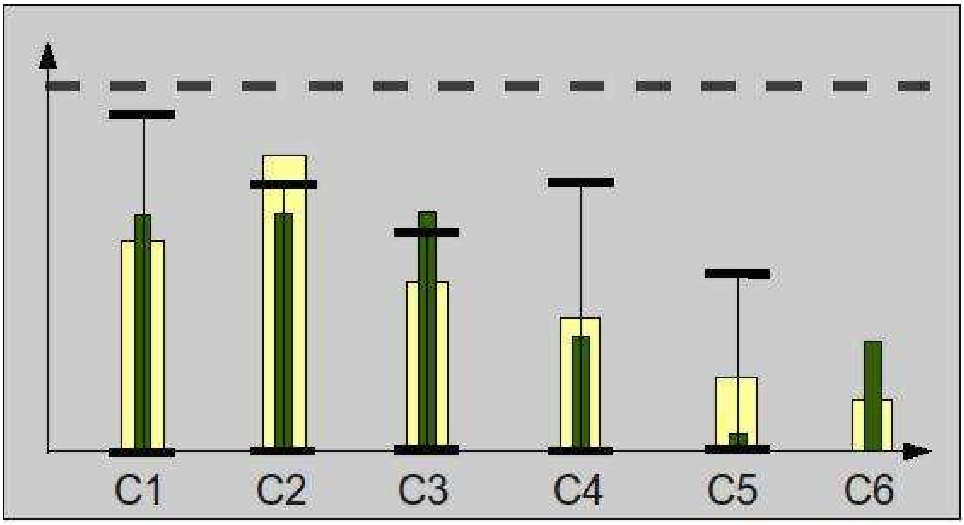

Figure 4.3: Knowledge level for a learner and class, as well expected knowledge level, in expanded form.

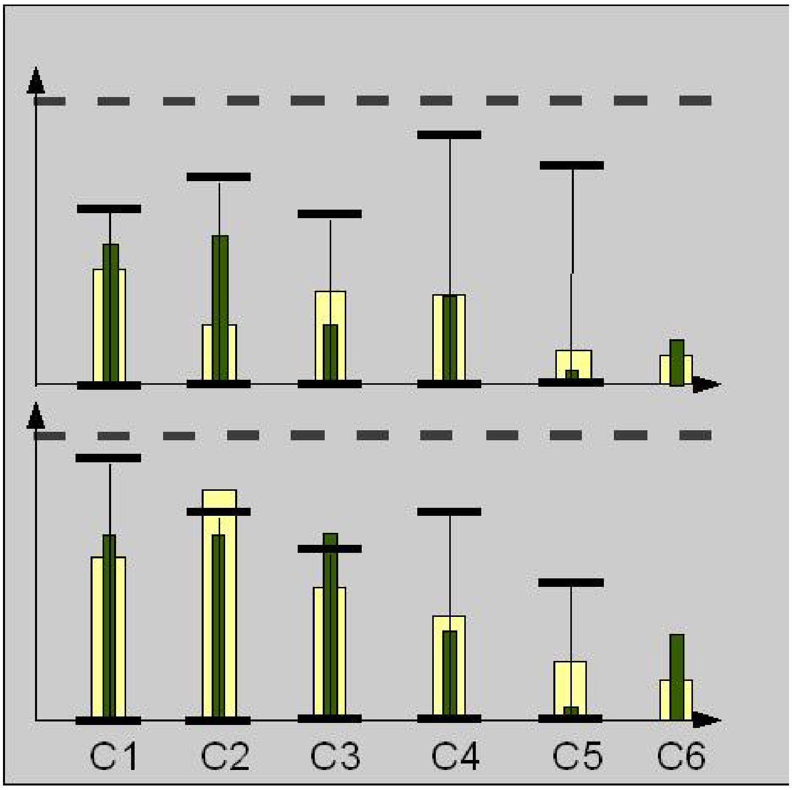

Figure 4.4: Knowledge level for a learner and class (with expected knowledge level) with history: the situation at a previous point in time and, at the bottom, at the present one.

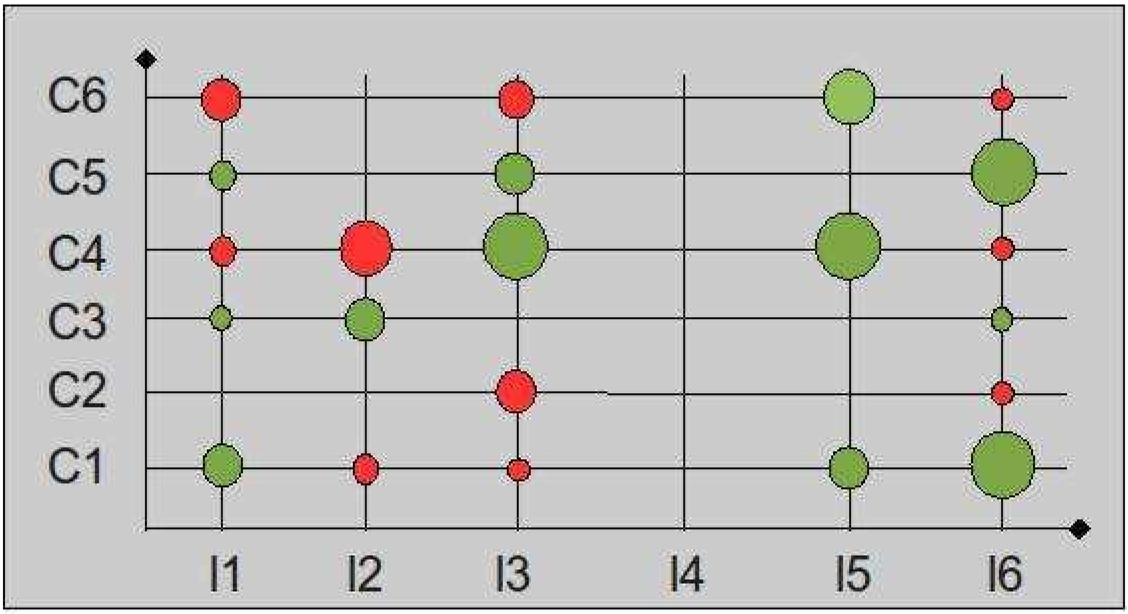

Figure 4.5: Knowledge level representation as matrix (learners - concepts).

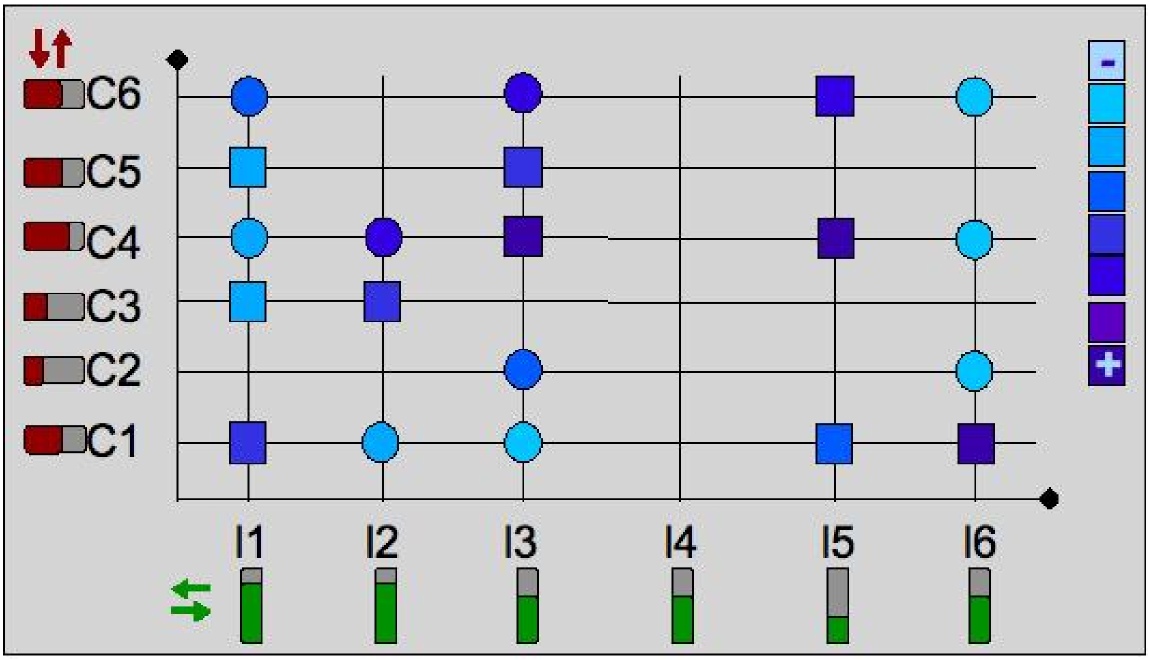

Figure 4.6: Another knowledge level representation as matrix (learners - concepts): here some average data is also reported for reference purposes.

learner (KnowStud). The arrows can be used to interact with the view to sort the columns or the rows in either ascending or descending order, according to the knowledge level of student per column, or the overall knowledge of the class per rows.

4.1.2.2 Activities

In our definitions, each structured interaction (normally called job) that a user has with the LMS is classified as an *Activity*: examples are filling a form, answering a question in a forum, uploading a document, contributing to a wiki and so on. Another interesting piece of information to be represented in a didactic environment is the activities performed by the learners. In fact, data about the number of activities performed encodes the active behavior of the user inside the system and, as an example, this can be used from tutors to get an indication about the evolution of the learner involvement and to check for possible discrepancies that can suggests an insufficient final knowledge level acquired by the learners.

4.1.2.2.1 Learner Views In this context, learners can be provided with the information about the activities visited. A look-and-feel similar to the previous visualisations was adopted. The following information is included:

- *ActStud*: Number of activities performed by a learner
- *ActClass*: Number of activities performed by the class
- *ActList*: Number of activities available in the course

Figure 4.7: Number of activities visited by a learner in textual form.

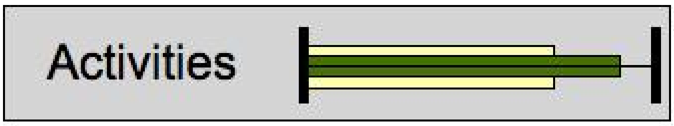

Figure 4.8: Number of activities visited by a learner in graphic form. The green bar represents the student data and the yellow bar represents the class data.

The following representation also includes the average value of activities visited by the students of the class (in yellow). These last two visualisations are intended to be available as additional widgets but also for direct integration with the LMS interface.

The learner, through the following representation, can also explore details about each single activity. The horizontal bar next to the activity name (in red) represents the percentage of learners that have accessed that activity, while the vertical bars represent two types of information: the blue one shows the percentage of the activities accessed by the learner, the green one the percentage of the activities accessed by the class. The learner can therefore compare his/her status with the overall class status. The bullet point indicates the fact that the current learner has visited the activity.

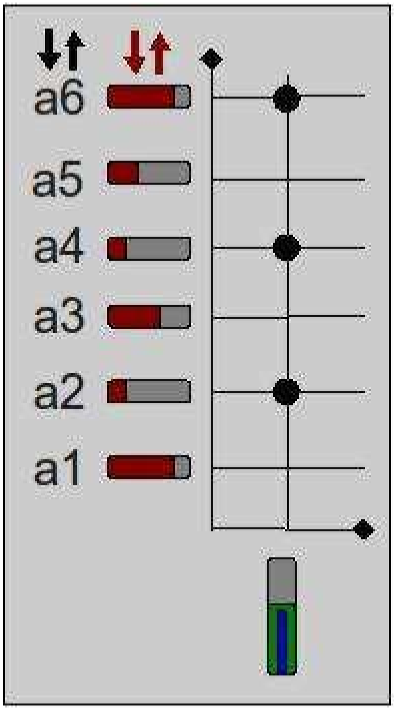

Figure 4.9: Analytic view of the resource viewed by a learner.

In Fig. 4.10, another analytical visualisation is presented. In this version, the bullet points is substituted by a completion flag that complements the average level of activities, represented in the main area of the view. The user can rearrange the presentation order simply by clicking on the appropriate small arrow on the top of the column. Another way to encode the number of views for each activity is by using the radius of the bullets or adding a new textual description.

4.1.2.2.2 Instructor/Tutor Views Instructors are supported by a view (Fig. 4.11) based on a matrix which represents the learners (or a selected subset of) in columns and the resources in rows: the filled circles indicate which resources were accessed by a specific learner. A possible

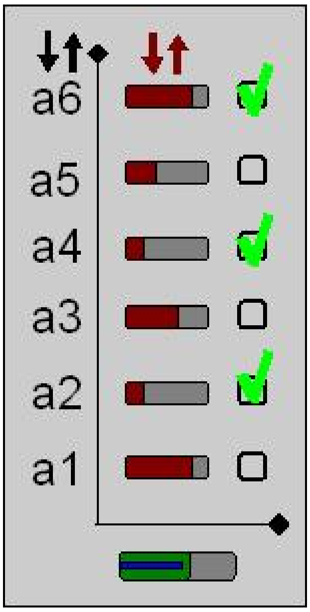

Figure 4.10: Alternative analytic view of the resource viewed by a learner.

variation could see the adoption of a scaling mechanism (with the same approach already seen in Fig. 4.5) to encode the number of times a learner has accessed a particular resource, as opposed to using a binary variable such as the circle presence/absence. The horizontal bar beside the activities name encodes the percentage of activities accessed by all students, while the vertical bar below the student name indicates the percentage of activities accessed by the student. The graph can be reorganised in either ascending or descending order –using the two arrows displayed next to the field– on the student names, activities name, values of the horizontal bars and of the vertical bars.

4.1.2.3 Goals

Learning goals are important indicators of the advancement of the learning process. They could be represented as a combination of learner characteristics (e.g. the level of knowledge achieved, the number of activities performed, etc). The list of goals has normally to be defined by the instructional designer in a dedicated tool using a specialised notation or language (which in the case of GRAPPLE is the CAM, with its graphical metaphor for an internal XML–based language). When the user reaches a particular goal, this information has to be detected and stored in the user model, allowing for its later usage. In the GRAPPLE case, this information is stored in the GUMF User Model in terms of a boolean flag that GVIS can access as a union (i.e.

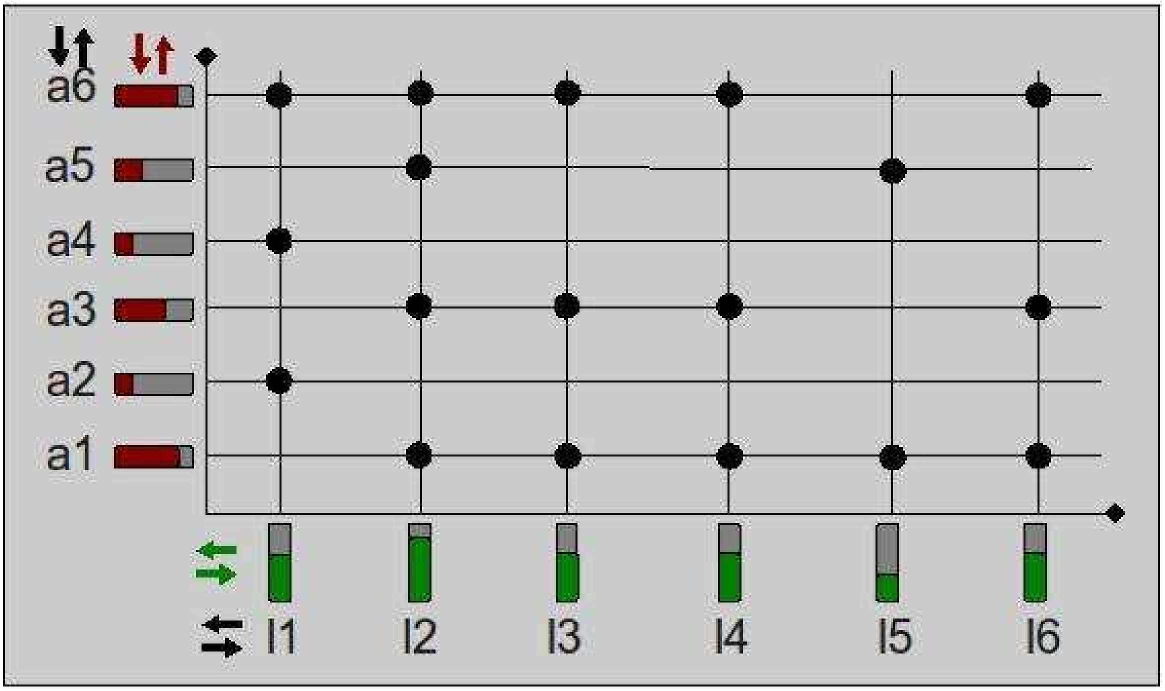

Figure 4.11: Details of resources visited by students.

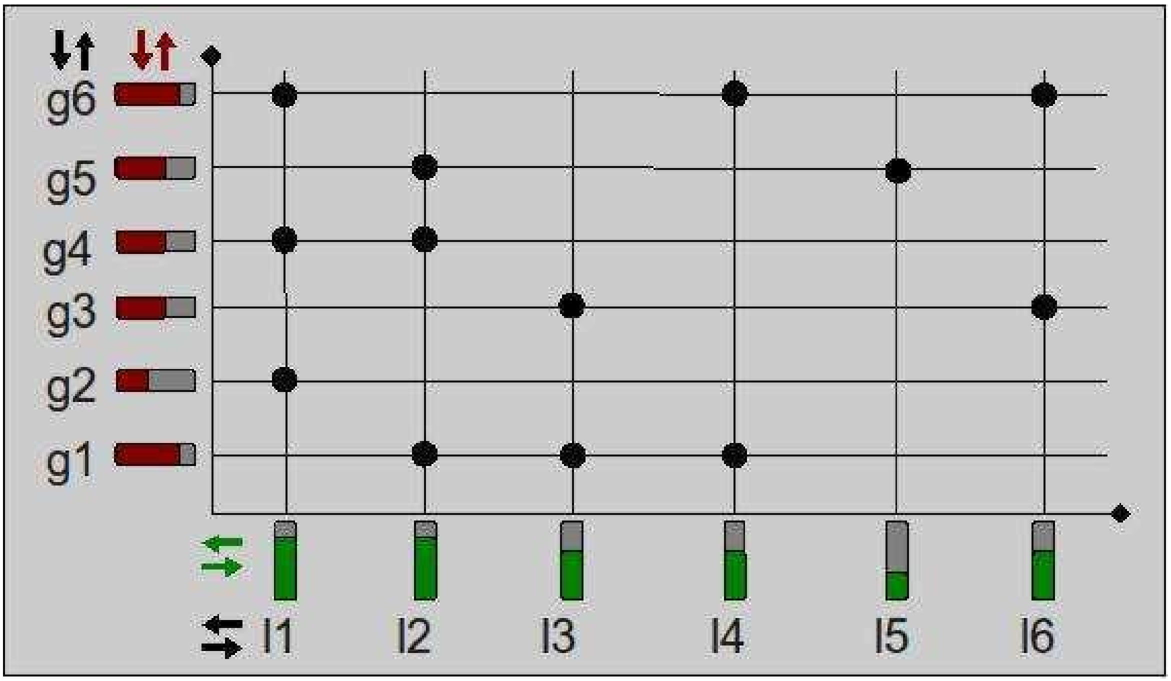

Figure 4.12: Goals achieved by learners.

join) between the didactic model (which contains the information about goals) and the user model (which encodes information about the learners activities). The information represented by the following views consists of:

- *GoalStudNum*: The number of learners that have achieved a particular learning goal
- *GoalStud*: Indication of whether a learner has achieved the learning goals

As previously stated, views are classified based on the stakeholder (i.e. consumer) they are primarily intended for.

4.1.2.3.1 Instructor/Tutor Views In the instructor/Tutor view a matrix of the goals achieved by the learners (see Fig. 4.12) is presented for awareness purposes and as also to serve as an analysis tool, to either identify problematic goals or problematic situations (e.g. a student failing on all the goals) or to show common patterns.

4.1.2.3.2 Learner Views The learner view presents a list of all the goals settled in the current course accompanied by the indications about which of them are achieved. This widget is another candidate to be also made available for direct inclusion in the LMS interface. The detailed view that will allow the learner to explore the analytical situation is presented in the left part of Figure 4.13. Like in the previous views (see Figg. 4.9 and 4.10), the bars represent aggregated data and work as references.

4.2 Evaluation of the Visualisation Mock-Ups

During the course of formative evaluation the visualisation ideas developed have been addressed in a survey. The visualisation mock-ups have been presented to a sample of users in order to gather feedback on the perceived benefits, usability and acceptance.

4.2.1 Introduction

The visualisation tools in GRAPPLE are intended for instructors as well as for learners.

Instructors are provided with different –and more detailed in respect of the learner's ones– visualisations to get an overview of the learning progress of a class or group of learners. The expected benefit is an empowerment of teaching by making the instructor aware of the different needs of different learners.

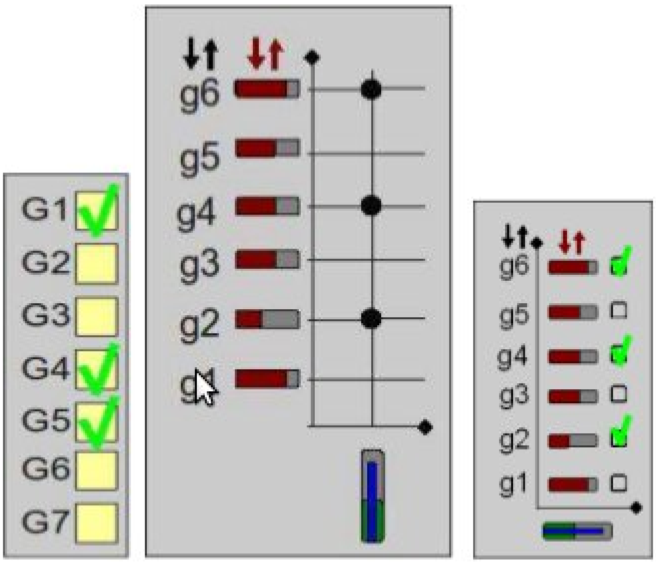

Figure 4.13: Goals achieved by the current learner. On the left: the representation of the compact indicator. On the center and on the right: two possible representations of the detailed views for the dashboard.

For learners a visual feedback on their own learning and progress as well as on where they stand in relation to their peers is important in order to promote reflection on their learning and enhance their knowledge awareness. The expected benefit is therefore empowerment in relation of learning effectiveness and of collaboration. Usability/suitability and acceptance are evaluation topics which are relevant for both profiles. Regarding the usability and suitability, crucial questions are whether the visualisations are suitable for their intended purpose and whether they are self-descriptive and understandable.

The objective of the survey conducted was to investigate whether the visualisation ideas feature an appropriate level on all these expected benefits and topics. These aspects could be investigated for visualisation mock-ups by gathering subjective assessments from users.

Instead of presenting all visualisations (i.e. for learners and for instructors) to all participants, it was deemed reasonable to create different paths of the survey according to the visualisation types. Consequently, learners should only be queried on visualisations intended for learners and instructors only on visualisations intended for instructors. Respondents who declared both profiles were offered the full set of questions.

The following sections outline the methods and outcomes defined for visualisation mock-ups evaluation.

4.2.2 Method

In this section information about participants and the materials used will be provided, including the visualisations and the three different questionnaires presented to learners, instructors and combined profiles.

4.2.2.1 Participants

A total of 45 participants took part in the mock-ups evaluation: 20 males and 25 females. Participants were recruited by the GRAPPLE project partners from academic environments of different backgrounds (mainly computer science and psychology). Participants were recruited in the third quarter of year 2008 from “Technical University of Eindhoven”, “Università della Svizzera italiana”, “Graz University”, “Open University Netherlands”, “Trinity College Dublin”, and other academic institutions.

Participants were 29.12 years old on average (SD = 6.0), with ages ranging between 22 and 46 years old.

Of the overall sample 13 people identified themselves as learners and filled out the questionnaire on the student visualisations only. 2 denoted themselves as instructors and filled out the part related to the instructor visualisations only. The rest (30 people) felts related to both roles and filled out the complete survey covering both types of visualisations (compare Table 4.7). This is connected with the fact that the first evaluation was run in a research context, with the majority of respondent interested in evaluating both the interfaces. This sample gave a total of 42 questionnaires compiled for the profile of *student* and 32 for the instructor ones, all of them valid, thanks to the fact that the interfaces for the distinct profiles proposed in the analysis were independent.

	Number	**Percentage**
Learner	13	28.9%
Instructor	2	4.4%
both roles	30	66.7%
Total	45	100.0%

Table 4.7: Sample of the participants at the visualisation evaluation

4.2.2.2 Material

This section will present the visualisations considered and the questionnaire used for the evaluation.

4.2.2.2.1 Visualisations For the purpose of the survey, a collection of visualisations was selected to be used. In order to make the survey concise and direct to the point, we presented a unique visualisation for each task identified, called primary. For measuring then the user considerations about any other variation in respect of the primary one for the task, we simply asked its perception of an higher level of helpfulness.

Visualisations for the learners For this part, three primary visualisations were addressed: **VisST1** (Fig. 4.2), that describes the comparison between knowledge on a single concept acquired by learner and class, **VisST2** (Fig. 4.3), that presents the knowledge level for each concept in the course for learner and class in respect of the expected level, and **VisST3** (Fig. 4.9), that represents the access to the activity for learner and class. Additionally, four possible variants of those visualisations were also presented. A question for each one of them on whether respondents find the specific variants helpful or preferable over the original one was included in the survey (see Table 4.8).

Variant	Question
Variant (I) of VisST1 (Fig. 4.7)	Would you find a similar visualisation of the number of activities (i.e. learning objects) visited by a learner helpful?
Variant (II) of VisST1 (Fig. 4.1)	Would you prefer a numerical representation over the graphic form?
Variant (I) of VisST3 (Fig. 4.10)	Would you prefer the following visualisation over the one presented above?
Variant (II) of VisST3 (Fig. 4.13, center)	Would you find a similar visualisation of the learning goals achieved by a learner helpful?

Table 4.8: Variants of the student visualisations.

Visualisations for instructors The instructor part of the survey included two visualisations (see Figg. 4.5 and 4.6) and a possible variant, shown in Fig. 4.11. VisI1 (Fig. 4.5) describes the knowledge level for the class. The visualisation is based on a matrix, where learners are mapped into columns and concepts into rows. The level of knowledge of a student on a concept is mapped into the dimension (in particular, the radius) of each circle of the matrix. The color in the circle represents whether the student reached (in green) or didn't reach (in red) the target knowledge level. A variation (Fig. 4.6) adopts a color scale to represent the level achieved by each learner on subjects and a shape to encode the reaching of the expected level – i.e. square for reached and round for unreached/under-expressed.

VisI2 (Fig. 4.11) shows the details of the resources visited by students. A matrix represents the learners (or a selected subset of them) in columns and the resources in rows: the filled circles indicate which resources a learner has accessed. The horizontal bars near the activity's name encode the percentage of the students who accessed the activity, while the vertical bars above the student's name indicate the percentage of activities accessed by the student.

Evaluation Questionnaires In order to potentially segment the set, a short demographic section had to be filled out by the participants to the questionnaire. In order to reduce the workload for the respondents and focus more clearly on the relevant part of giving feedback on the visualisations, the section of the questionnaire about demographic data was designed to only include questions on gender, age and recruiting project partner.

The questionnaire for Subjective Assessment of Visualisation used in the survey was based on the Evaluation Guidelines elaborated for GRAPPLE [1], with some small adaptations. The minor changes related to the explicit distinction between the different visualisation types with the aim of reducing the workload for participants.

The questionnaire consisted of three main parts referring to usability/suitability, visualisation benefits and acceptance. While the first two areas are covered using lower granularity indicators, identified in the next list, for the acceptance section a direct approach was adopted and the participants' opinion was gathered using a two items global indicator.

- Perceived usability/suitability - in terms of:
  - **suitability for the task**
  - **self-descriptiveness**
- Visualization benefits :
  - **for student visualisations** :

[1] the GRAPPLE Evaluation Guidelines is the internal document elaborated by UniGraz to guide the interview and data collection phases for all the project.

  * *Meta-cognition*
  * *Cognitive load*
  * *Learning effectiveness*
  * *Benefits for peers/collaboration*
- **for instructor visualisations** :
  * *Meta-cognition*
  * *Cognitive load*
  * *Benefits for instructors (personalised/individualised instruction)*
  * *Acceptance*

Each aspect of the sub-scales was assessed through two questions, i.e. statements which had to be answered on a five-point *Likert* rating scale with the extreme values as “strongly disagree” (=1) and “strongly agree” (=5). This way, a mean score averaging across the two items could be calculated for each aspect. Furthermore, an overall usability score could be calculated from the usability scores for the two aspects of usability. In order to gather information for deriving of ideas on how to improve the visualisations and their benefit for learners and instructors, one open question was included for each visualisation in the questionnaire (i.e. *Could you please describe in detail your opinion on this visualisation (i.e. positive/negative aspects, how to improve it)?*).

The questionnaire was implemented and submitted as an online survey using the LimeSurvey tool[1]. The three different branches of the questionnaire corresponding to the perceived role in the educational process (learner, instructor, or both) were realised through the application of a conditional question. As a result, depending on his/her answer a respondent was presented either with the visualisations for learners, for instructors or both[2].

4.2.2.3 Procedure

The first evaluation of GRAPPLE visualisations was carried out in March 2010. The procedure was as follows:

(a) First of all, the demographic questionnaire had to be filled out.

[1] for more informations please refer to the deliverable D8.1b - ”Refinement and Improvement of Evaluation Framework” in the section 2.1 - Online Survey Tools on the GRAPPLE website (Steiner and Hillemann 2010).

[2] The complete questionnaire for both student visualisations and instructor visualisations can be found in the appendix of the deliverable D4.5c (Mazza, Mazzola, Glahn, Nussbaumer, and Verpoorten 2011)

(b) This was followed by a short introduction of the goal of the user survey, which ended with a question on the perceived role. Depending on the respondents answer to this question the relevant type of visualisation was presented (i.e. learner, instructor/teacher, or both).

(c) Participants were then presented with the first visualisation to be evaluated, which was accompanied by the explanation around its purpose and other information.

(d) Participants were requested to have a close look at each visualisation and to subsequently fill out the questionnaire for the respective visualisation.

(e) *[if available]* They had to provide answers to the questions on variants of the last shown visualisation.

(f) For the different visualisations to be evaluated, steps (c), (d) and (e) were repeated for each visualisation. Subsequent to giving feedback on one visualisation, the next visualisation was provided along with the questionnaires (if available) and answered by the participants, and so forth.

4.2.3 Results

As previously stated, the possible score for each question spanned between 1 and 5, with higher values indicating a better result. Values above the middle of the scale were considered, in accordance with the guidelines defined in the project, as rather good (with higher values indicating a better result) and values in the lower half of the scale were seen as an indicator of a low quality of the visualisation.

In the following section, the results for the student visualizations and their variants are first presented. Subsequently, the results for the instructor visualisations are described. According to the branched querying mechanism, 43 participants gave feedback on the student visualisations, while the instructor visualisations were assessed by a total of 32 participants.

4.2.3.1 Results for student visualisations

For VisST1 (see Fig. 4.2) there are medium to good results in all aspects. The average scores range from 2.95 (SD = 0.73) in the cognitive load scale to 4.14 (SD = 1.04) in the self-descriptiveness scale. The best results could therefore be found in the general usability scale (M = 4.06; SD = 0.77) and its two sub-scales suitability for the task (M = 3.98; SD = 0.76) and self-descriptiveness (M = 4.14, SD = 1.04). Table 4.9 below shows all results in details.

Sub-scales	**Nr.**	**Min.**	**Max.**	**Mean**	**S.D.**
suitability for the task	43	1.00	5.00	3.98	0.76
self-descriptiveness	43	1.00	5.00	4.14	1.04
usability	43	1.00	5.00	4.06	0.77
metacognition	43	2.50	5.00	3.62	0.69
cognitive load	43	1.00	4.50	2.95	0.73
learning-effectiveness	43	1.00	4.50	3.34	0.84
benefits for peers	43	1.00	5.00	3.42	1.09
acceptance	42	1.00	5.00	3.30	1.16

Table 4.9: Results of the evaluation for the VisST1 widget.

Most respondents (24 out of 42, i.e. 53.3%) found a similar visualisation of the number of activities visited (see Fig. 4.7) helpful; however, 8.9 % (4 out of 42) of the respondents expressed the opposite feeling. The other 14 respondents (31,1%) did not have an opinion or preference on the matter. The majority of respondents i.e. 29 out of 43 people (64,4%), would not prefer a numerical presentation of the knowledge acquired (see Fig. 4.1) over the graphical form. On the other hand, 13 people (28,9%) would prefer this numerical kind of presentation.

The answers to all open questions were analysed in terms of contents in order to identify some general categories reflecting the participants answers and to be able to calculate its recurrence. Thereby, answer categories were distinguished referring to either positive aspects, negative aspects and issues for improvement. The results on the open question for VisST1 (*Could you please describe in more detail your opinion on this visualisation (i.e. positive/negative aspects, how to improve it)?*) are summarised in Table 4.10.

The most prominent feedback was that respondents answered that this visualisation provides good and useful information (8 people) and it is clear and easy to understand (6 people). Also, the motivating effect through the comparison with peers was mentioned (5 people).

On the other hand, the comparison to the class was indicated as possibly problematic as it might negatively affect self-esteem and collaboration (7 people). Upon further analysis of the answers provided, some of them thought that, specifically with regards to underachievers, this kind of comparison could be disadvantageous.

The fact that this visualisation features no automatic scaling and provides no explicit reference space was criticised (respectively 5 people and 5 people). Therefore, in order to better understand the scale it was deemed desirable to add scaling and to label the bars (3 people).

The real capacity of this view to support the learning was questioned by 5 people whereas

Positive aspects	
Category	**Number**
good/useful information	8
Clear and easy to understand	6
Motivating effect through comparison to the class	5
Support self-awareness/realism	2
Negative aspects	
Category	**Number**
Possible neg effects on self-worth or on collaboration different visualisations for underachievers	7
No scaling, add scaling	5
Reference space/context unclear and information therefore not meaningful	5
Overlapping of bars	5
Support of learning questionable	4
Very broad information more details would be desirable	4
Not self-descriptive	2
Issues for improvements	
Category	**Number**
Labeling of bars	3
Improve graphic style, colors	2
Bars should be vertical instead of horizontal	2
Possible pos/neg effects should be investigated	1

Table 4.10: Qualitative feedback concerning VisST1.

2 people thought that it is able to support self-awareness and realism in the learner. Some respondents (4 people) suggested to concentrate more on the information represented – or provide more information – and 2 people also deemed the information as non self-descriptive. Finally, suggestions to improve graphic style and colors was reported (2 people), along with an indication to tilt bars from a horizontal to a vertical position (2) and one person requested to better investigate the trade–off between positive and negative effects.

Similarly to the first visualisation, the second mock-up for the student visualisation (VisST2, see Fig. 4.3) also received medium to good results in all aspects (compare Table 4.11). On the whole, however, most mean scores lie slightly below the results for VisST1. The scores range from 2.89 (SD = 0.84) for cognitive-load to 3.83 (SD = 1.08) for suitability for the task. Similarly to VisST1, the highest scores were reached on the usability sub-scales suitability for the task with value 3.83 (SD = 1.08) and self-descriptiveness with value 3.61 (SD = 1.27). The average usability score is 3.73 (SD = 1.06).

Subscales	Nr.	Min.	Max.	Mean	S.D.
suitability for the task	38	1.50	5.00	3.83	1.08
self-descriptiveness	38	1.00	5.00	3.61	1.27
usability	38	1.25	5.00	3.73	1.06
metacognition	38	1.50	5.00	3.63	0.91
cognitive load	38	1.00	4.50	2.89	0.84
learning-effectiveness	38	1.50	5.00	3.60	0.86
benefits for peers	38	1.50	5.00	3.41	1.00
acceptance	37	1.00	5.00	3.12	1.31

Table 4.11: Results of the evaluation for the VisST2 widget.

With respect to this visualisation a question was also included on whether a similar variant showing the number of activities visited would be helpful. The participants opinion on this visualisation variant on the number of activities visited was not that clear. Still, 17 out of the 39 people (37.8%) that answered this question indicated such a visualisation as being helpful, 10 people (22.2%) did not think so and 12 people (26.7%) did not have an opinion.

The results on the qualitative feedback gathered through the open question are presented in Table 4.12. The most frequent answer (11 people) was that this visualisation provides useful information, concerning the relation to the maximum knowledge achievable and the learning progress and 4 people appreciated the simplicity and clarity of the visualisation.

On the other hand, respondents indicated that the information is arranged and shown in a very complex way and is therefore hard to interpret and understand (7 persons).

Moreover, the possible negative effect on self-esteem due to the direct comparison to the class was mentioned once again. Participants would like to see the graphic and color of the visualisation improved and they would add scaling and explanations (2 people for 2 people respectively).

Another reported substantial aspect was the visualisation positive effect on improving the learning through awareness of the goals (2 people), even if one of them reported that the value for learning is questionable and the one that there is lack of details for this objective to be reached.

Critical comments were provided about the difficult to interpret the meaning of the horizontal lines, on the overlapping bars and on the unclear labeling of the axis (respectively 4, 1 and 1 people).

The presence of enough information was appreciated by 2 people and the compactness of the visualisation by another one. Finally, some suggestions for improvements were about providing explanations, enhancements of labeling and description in the visualisation. A person suggested that no contextual information from the class should be provided to the learner for the maximum efficacy.

Positive aspects	
Category	**Number**
Useful information in relation to max. knowledge, progress	11
Simplicity, clarity	4
Awareness of goals may enhance learning	2
Differentiation, more details	2
Lots of information in a compact representation form	1
Negative aspects	
Category	**Number**
Complexity, hard to interpret/understand	7
Possible neg. effect on self-worth	5
Horizontal lines hard to interpret/understand	4
Lack of details	1
Value for learning questionable	1
Overlapping bars	1
Axes not labeled - unclear	1
Issues for improvements	
Category	**Number**
Improve graphic and color	2
Add scaling	2
Explanations important	2
Include labeling/description	1
Only individual learning performance should be visible	1

Table 4.12: Qualitative feedback concerning VisST2.

The results for the student visualisation 3 (VisST3, see Fig. 4.9) depicted a different picture than those for the two previous visualisations. The evaluation resulted in rather low to medium average scores on all aspects (see Table 4.13). The least favorable results (i.e. lowest scores) could be found for self-descriptiveness with 2.06 (SD = 0.93) and user-acceptance with 2.08 (SD = 1.06). The best result was reached on the sub-scale benefits for peers with a score of 3.06 (SD = 0.80).

The visualisation variant of this mock-up representing the activities visited by a check mark

Subscales	Nr.	Min.	Max.	Mean	S.D.
suitability for the task	40	1.00	4.50	2.55	1.07
self-descriptiveness	40	1.00	4.00	2.06	0.93
usability	40	1.00	4.00	2.30	0.95
metacognition	40	1.50	5.00	2.84	0.80
cognitive load	40	1.00	4.00	2.24	0.78
learning-effectiveness	39	1.00	4.50	2.82	0.87
benefits for peers	39	2.00	4.50	3.06	0.80
acceptance	38	1.00	5.00	2.08	1.06

Table 4.13: Results of the evaluation for the VisST3 widget.

(see Figure 4.10) would be preferred by the majority of the respondents (32 out of 41, i.e. 71.1%); only 5 respondents (11.1%) expressed the opposite opinion. A similar visualisation of the learning goals achieved by a learner (see the right part of Figure 4.13) was found helpful by 15 people out of 39 (38.5%). 10 respondents (25.6%) did not agreed and 14 (35.9%) did not express any opinion. The results for the qualitative feedback are summarised in Table 4.14. Contrary to VisST1 and VisST2, many of the participants attribute the complexity of this third visualisation to too much information covered (17 people). In line with this, some of the respondents explicitly expressed the wish for more simplicity (3 people).

A comparison between the student visualisations showed that for all aspects both VisST1 and VisST2 achieve better results than VisST3. Only for the sub-scale benefit for peers no significant differences could be identified, as can be clearly seen in Table 4.15.

4.2.3.2 Results for Instructor Visualisations

The Instructor visualisation mock-up VisI1 (see Figure 4.5) received medium to good results in all aspects (see Table 4.16). The lowest average score was 2.86 (SD 0,80) for cognitive load. The best result was on the benefit for instructors aspect with a mean score of 3.74 (SD 0.93).

Regarding the qualitative feedback (see Table 4.17) most of the respondents think that this visualisation is clear and useful (10 people), although 5 persons also indicated that they find it too complex and therefore difficult to understand. The use of the circle dimension to indicate the knowledge level was considered problematic by 3 respondents. Suggestions around improvements of the visualisation mentioned were the improvement of the general style of the visualisation, as well as the realisation of additional explanations and last but not least, more simplicity (2 answers and 2 answers respectively).

Positive aspects	
Category	**Number**
Depicting information more objectively	2
Motivating effect	1
Overview on a lot of information at one time	1
Negative aspects	
Category	**Number**
Complexity, too much information	17
Not-self-descriptive	3
Horizontal and vertical bars are confusing	2
Not visually appealing	2
Negative effect of comparison to others	2
Comparison to others on activity level not necessary	1
Uses visual guitar metaphor but is not explainable this way	1
Arrows not understandable	1
Issues for improvements	
Category	**Number**
Simplify	3
Add text/explanation	1
Different symbol for visualising visited activities	1
Use different (better distinguishable) colors	1
General remark: the other visualisations provide better information/comparison	1

Table 4.14: Qualitative feedback concerning VisST3.

Two respondents liked the visibility of the learning progress and 2 more reported a preference for this one over VisST2 and VisST3 (despite the fact they are intended for different profiles, so not meant to be compared), whereas 2 people reported that the absence of explicit information in the visualisation could be problematic. Finally, one person indicated that the labellings of the axis could improve the widget.

Regarding the instructor visualisation VisI2 (see Fig. 4.11) we registered average scores ranging from 2.57 (SD = 0.75) on the cognitive load scale to 3.28 (SD = 0.82) on the benefit for instructors scale. This can indicate a medium to low quality of the visualisation. The mean scores on all aspects are lower compared to the first instructor visualisation. The best result was reached on the benefit for instruction (3,28, SD 0,82). The results on all sub-scales can be found in Table 4.18.

The qualitative analysis showed that the 24.4% of the participants (11 out of 26) would find

	VisST1			VisST2			VisST3		
Subscales	**NR.**	**Mean**	**S.D.**	**NR.**	**Mean**	**S.D.**	**NR.**	**Mean**	**S.D.**
suitability for the task	43	3.98	0.76	38	3.83	1.08	40	2.55	1.07
self-descriptiveness	43	4.14	1.04	38	3.61	1.27	40	2.06	0.93
usability	43	4.06	0.77	38	3.73	1.06	40	2.30	0.95
metacognition	43	3.62	0.69	38	3.63	0.91	40	2.84	0.80
cognitive load	43	2.95	0.73	38	2.89	0.84	40	2.24	0.78
learning-effectiveness	43	3.34	0.84	38	3.60	0.86	39	2.82	0.87
benefits for peers	43	3.42	1.09	38	3.41	1.00	39	3.06	0.80
acceptance	42	3.30	1.16	37	3.12	1.31	38	2.08	1.06

Table 4.15: Comparison of the results of the evaluation amongst the widgets VisST1, VisST2, and VisST3.

Subscales	Nr.	Min.	Max.	Mean	S.D.
suitability for the task	29	1.00	5.00	3.62	1.01
self-descriptiveness	29	1.00	5.00	3.48	1.18
usability	29	1.25	5.00	3.55	1.02
metacognition	29	2.00	5.00	3.50	0.79
cognitive load	29	1.00	4.00	2.86	0.80
benefits for instructors	29	2.00	5.00	3.74	0.93
acceptance	28	1.00	5.00	3.41	1.41

Table 4.16: Results of the evaluation for the VisI1 widget.

a similar visualisation for the learning goals achieved (compare 4.12) helpful; 17.8% (8 out of 26) did not agree on that and 15.6% (7 out of 26) did not express an opinion. Noticeable for this question was the low response rate – 42.2% (19 out of 45) – meaning that the majority of the participants did not provide any answer.

In the qualitative feedback almost half of the responding instructors (9 people) deemed this visualisation as too complex. They would improve the style and make it more up to date and less technical (4 people). Only two of the respondents explicitly stated that they found this visualisation useful. Some respondents indicated that the visualisation is not appealing (4 people). Another critical point indicated by a single person was about the lack of legend and explanations, and the difficulty in understanding the function of the arrows.

Someone declared (as already stated for VisI1) that the simplest visualizations i.e. VisST1 and VisST2 were preferable; some improvements suggestions consisted in offering a 2-step overview with the possibility to zoom in and simplify the view (1 person and 1 person respec-

Positive aspects	
Category	**Number**
Clear and useful	10
Learning progress visible	2
Preferred over VisST2 and VisST3	2
Negative aspects	
Category	**Number**
Too complex, difficult to understand	5
Circle dimension for depicting knowledge level not optimal	3
No explanation	2
Issues for improvements	
Category	**Number**
Improve style (web2.0, 3d-model)	2
Add explanation/text	2
More simplicity	2
Add axes labels	1

Table 4.17: Qualitative feedback concerning VisI1.

tively).

For more detailed information about the answers received refer to Table 4.19.

A comparison between the instructor visualisations suggested that there was no important difference in any aspect between the two visualisations, rather just a generalised preference for VisI1.

4.2.4 Discussion on the evaluation of mock-ups

The evaluation results for the first two student visualisations indicate a medium to good quality of visualisation in respect to all the aspects for both VisST1 and VisST2.

The aspect of *usability/suitability* – with its two sub-scales *suitability for the task* and *self-descriptiveness* – reached the highest values. This result suggests that these visualisations are suitable for their intended purpose and also largely self-descriptive and understandable. Learners think that this visualisation is suitable for getting an overview of the current status in the learning process. According to the participants the qualitative feedback for the two visualisations is easy to understand and not unnecessarily complex.

With regards to the third student visualisation, VisST3, the results are significantly lower than those obtained for the two other visualisations. The reason for this could be that this

Subscales	Nr.	Min.	Max.	Mean	S.D.
suitability for the task	29	1.00	5.00	3.00	0.94
self-descriptiveness	29	1.00	5.00	2.66	1.22
usability	29	1.25	4.75	2.83	0.94
metacognition	29	1.50	5.00	3.17	0.88
cognitive load	29	1.00	4.00	2.57	0.75
benefits for instructors	29	1.50	5.00	3.28	0.82
acceptance	28	1.00	5.00	2.46	1.29

Table 4.18: Results of the evaluation for the VisI2 widget.

Positive aspects	
Category	**Number**
Useful	2
Negative aspects	
Category	**Number**
Too complex	9
Not appealing	4
Arrows not understandable	1
VisSt1 and VisST2 would be preferred	1
No legend/explanation	1
Issues for improvements	
Category	**Number**
Improve style (more up to date, less technical and cold)	4
2-step overview with the possibility to zoom in	1
Simplify	1

Table 4.19: Qualitative feedback concerning VisI2.

visualisation covers a lot of information and is not as simple as the previous two. This fact led to the desire for more simplicity by the participants.

For the two student visualisations VisST1 and VisST2 the results on *visualisation benefits* also indicated a medium to good quality. On the one hand learners think that the visualisations are able to help learners reflect on their own learning progress and accomplish their goals. On the other hand, the visualisations are also assumed to help learners to better understand their learning through comparison with other learners.

Some respondents, however, indicated that the comparison with the class might be problematic and negatively affect self-worth and collaboration. A few respondents hereby indicated that, specifically for underachievers, this kind of comparison could be disadvantageous. They suggested that it would be more desirable to offer a comparison with one self own prior performance only. This aspect would require more studies to be assessed.

On the instructor visualisations, VisI1 reached medium to good results in all aspects. VisI2 reached lower results in all aspects. While the most prominent qualitative feedback for VisI1 was that it is clear and useful, for VisI2 it was especially emphasised that its visualisation is too complex. Nevertheless, when it come to these instructor visualisations, respondents think that – in spite of their complexity – the visualisations are able to allow instructors to adapt their teaching to their students peculiarities by better understanding the students' needs. In fact, both visualisations achieved the best result on the relevant sub-scale i.e. *benefit for instructor*.

For VisI2 and VisST3, which use a similar representation for either an individual or a whole group of learners, the evaluation outcomes were similar. Both visualisations – who adopt a representation paradigm based on a matrix for depicting the activities visited and some contextual additional bars – received the least favorable results in their groups. These visualisations are probably perceived as representing too much information at a time.

In order to improve the *self–descriptiveness* of these complex visualisations some explanations or a legend describing the single components of the visualisation (e.g. the arrows in VisST3) should be added. Furthermore, the graphical style should be improved, for example by integrating different and better distinguishable colors. This way not only the *usability* of such rather complex visualisations could be enhanced but their *acceptance* could also be improved. *User acceptance* ratings for the current version of VisI2 and VisST3 were rather low, while acceptance scores for the other visualisations (i.e. VisI1, VisST1, and VisST3) were of medium to good quality. Both students and instructors, would make use of these visualisations in their work and they would also recommend them to others.

On the whole, the simpler visualisations were better perceived (e.g. providing useful information and an overview as well as being clear and easy to understand) while the visualisations covering a lot of information were stigmatised as complex and covering too much information and led to the wish for more simplicity. Both learners and instructors expressed the wish to improve the design and scrutability of the visualisations (e.g. integrating different and better distinguishable colors, adding more explanations and navigational facilities). Labeling or adding legends in the visualisation would be useful in order to improve the self-descriptiveness.

The aim of this first evaluation of the visualisation mock ups was to gather initial information through subjective assessment regarding usability aspects, visualisation quality and users acceptance of the visualisations. The evaluation results were used, among other things, to drive further improvements of the visualisation tools in their final implementation. In fact, the visualisations implemented and used in the project, and also in the experiments outside GRAPPLE presented in the next chapter, were selected amongst the higher rated ones or improved starting from them and using the received suggestions and feedbacks. Examples of these improvements are the simplification of the visualizations automatically provided to the user in the LMS interface, the possibility for autonomous exploration by the single learner of more analytical ones on needs, the choice of the class as the natural reference for the learner current status, and so forth. This approach was expected to maximise the benefit brought by their integration and use in an adaptive learning environment.

A further step of evaluation was then carried out based on the current implementation of the visualisation tools in other contexts than the GRAPPLE system, as will be presented in the next chapter. This also allowed to get a better understanding around additional qualitative aspects of the visualisations (e.g. actual learning performance, meta-cognitive abilities by contrasting objective and subjective learning outcome, perceived cognitive load).

Chapter 5

GVIS: experiments outside GRAPPLE

In this chapter the experiments conducted outside GRAPPLE are presented. In its first part a few experiments are described which are more directly related to LMS and ITS, whereas in the following section a number of cases with data integration occurring from outside a learning platform are detailed.

5.1 GVIS connected to LMS/ITS

A first application was developed on a Master course which was offered in blended mode. The GVIS application was integrated into the Moodle Learning Management System (see Fig. 5.1): it is the bottom block on the right-hand side column.

5.1.1 An Experiment with the Moodle LMS

The initial pilot was conducted by applying the GVIS architecture to the "Educational Communication and eLearning" course held during the winter term of the 2009-2010 academic year by prof. Cantoni at the Politecnico di Milano, in Italy, as part of the MS degree in Engineering in Computer Science (see Fig. 5.1).

Although for the purpose of this test we did not aggregate data from different sources, our visualisation infrastructure facilitated the aggregation of data coming from different tables in the Moodle database as well as the visual represention of contextual information about both the course and the learners. On top of the above, we developed a specific widget for representing the number of logins and the messages posted in a forum. An interesting outcome was the graphical

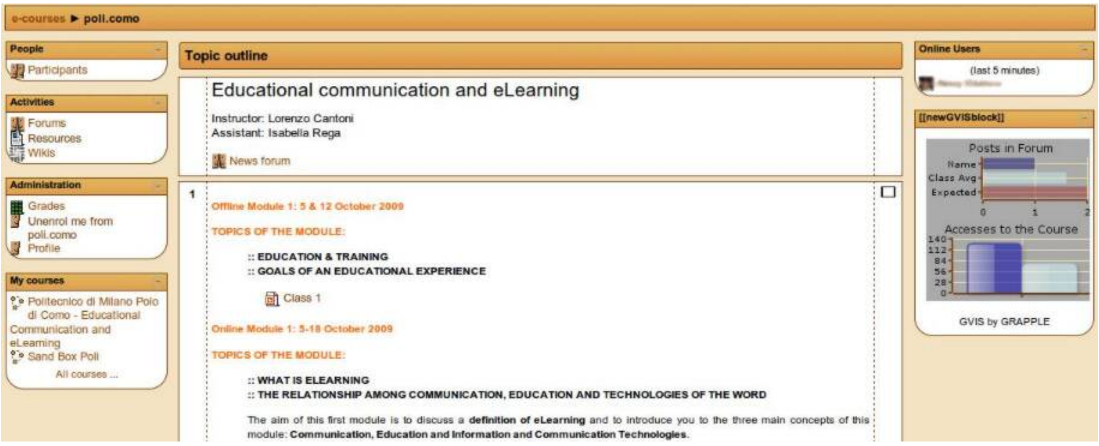

Figure 5.1: The output of the GVIS module when plugged into a Moodle course (the bottom block on the right-hand side column).

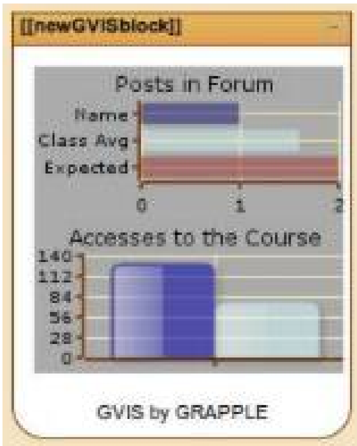

Figure 5.2: The output of GVIS module: numbers of logins and forum contributions.

comparison between the learner's specific information and the average value achieved by the class, which can work as reference for the self-monitoring process of the user's progress (Woolf 2009). We also implemented functionalities to support the learner: Fig. 5.2 shows the number of messages posted onto the forums by the current user as well as the count of the accesses to resources by the student throughout the course. The widgets also allows for the comparison with the class average and, in the first case, with the level expected by instructor as well. Both these pieces of information are considered important by the teacher assistant, who in this case also played the role of instructional designer and developed – relying of some technical support– the online part of the course. All the widgets were implemented on the basis of the requests and suggestions that came from the instructor and the teacher assistant. Others widgets that were developed for this pilot study are the number of logins onto the course grouped by date and by student as well as the number of forum entries posted by date (grouped by the number of people that posted each day, the total number of posts per day and per single student).

Fig. 5.2 shows a mix of accesses to the course/resources done by students as well as forum posts, which are the activities that are considered to be important by the Instructional designer, who developed the online part of the course. As it can be seen, the result is a graphical comparisons between the learner's specific information and the average value achieved by the class, which can work as a contextual reference to monitor the user's progress.

Some functionalities for the teacher assistant have been implemented in order to support the tutoring, as shown in Fig. 5.3, where a collage of two of the widgets provided is reported.

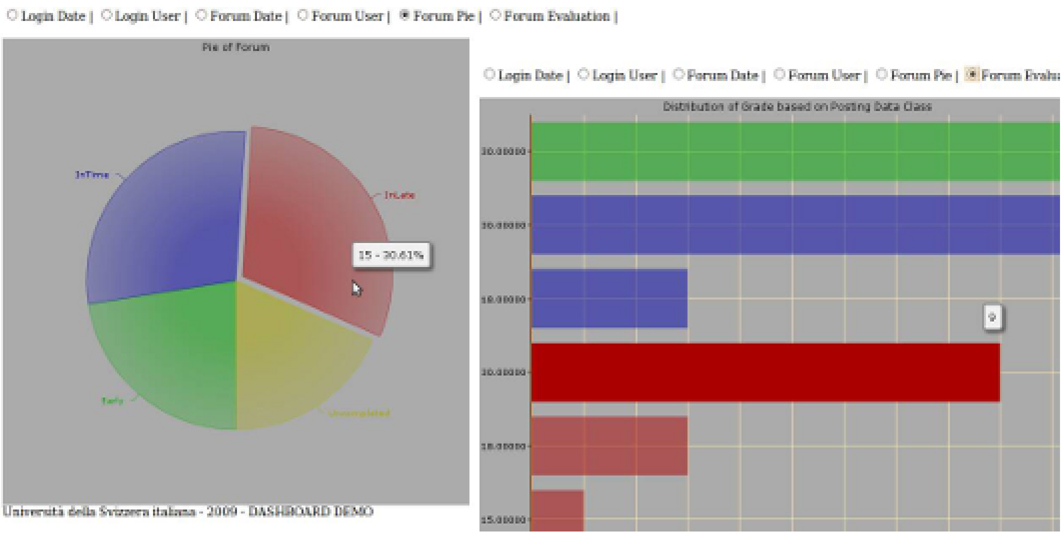

Figure 5.3: The GVIS module for instructor. A classification of the posts based on the submission date - Early, OnTime, Late and Uncompleted - is presented (on the left), accompanied by the related evaluation (on the right).

More specifically (see Fig. 5.3), groups of messages posted in the same date ranges and based on different deadlines, are depicted using different colors in the pie chart. Each slice of the pie represents a category based on the posting date with respect to the deadline set by the instructor. This can be: early, on time, late, or uncompleted. The size of the slice indicates the number of messages into each posting category. The instructor may assign a grade to each of the messages posted in the forums, depicted on the right-hand side of the bar chart. In this visualisation, bars represent the distribution of grades given by the instructor to the

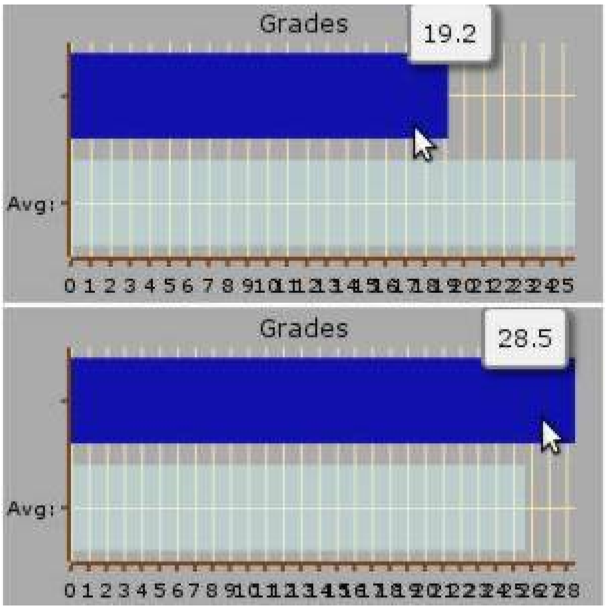

Figure 5.4: The graph of the grades for two different students from two distinct classes.

postings, and the color of bars represent the posting specific category (e.g. early, on time, late, or uncompleted).

5.1.1.1 The "grade" widget

In the online course there are different activities that have to be performed by the learners over specific intervals of time which have to be graded by the tutor throughout the semester.

For this reason a specific widget was developed to present the individual state as the average grade achieved by a student until that moment. This information, specific to every learner, is compared with the overall average grade of the class. This is particularly important because the student can autonomously check his current average grade at any point in time and not only at the end of the course. Because of the above, this can be safely considered as an awareness tool (Romero, Tricot, and Mariné 2009).

In Fig. 5.4 two screenshots are presented which refer to different learners in distinct classes and that are useful to make the students aware as to whether they are aligned with the rest of their colleagues.

5.1.1.2 Evaluation

An initial evaluation that allowed the learners to provide feedback on their experience with the system was implemented. This was composed by a questionnaire containing 16 questions on a 5-point based Likert-scale and it was submitted at the end of the course as an online survey. The possible answers ranged from complete disagreement (1 - SD) to full agreement (5 - SA) with the concept expressed.

The first interesting result was that 22 students out of 45 answered within the time-frame allocated. The first impression seemed promising, even though some experiments with a higher number of students and longer time-frame would be certainly needed to confirm these initial results, due to the very limited impact of the current one (i.e. only one course during a single semester).

Based on their very small number and the impossibility to group them coherently, it was not thought meaningful to include the non-quantitative answers received. After providing some introductory information about the aims and the scope of the survey, the questionnaire collected demographic data, such as age, gender and role of the respondents. Unfortunately, probably due to the context of the course and the predominance of male students (36 over a total of 45, means 75%), we collected no feedback from female ones. The average age value is 23.77 - with variation from a minimum of 22 to a maximum of 32 - and all of them compiled the survey as learners.

Table 5.1 reports an analysis of the results. For ease of interpretation we converted all the answers into a positive scale i.e. we inverted the scale for those questions that were asked in a negative form (i.e. marked by an asterisk). This process also allowed to group variables in order to perform a number analysis on them. The resulting sets were named respectively C1, C2, C3 and C4 based on the percentage of positive versus negative answers provided and on the standard distribution measure.

In the first group, the most positive one C1, was collected the questions USD1, USD2, and VBBI1 that are related to the easiness of understanding the visualizations, and to its capability of helping instructor in tailoring the teaching to individual learners.

In set named C2 - composed by the questions called UST1, UST2, VBM1, VBLE1, VBBP2 - the rate of positive answers is predominant, however some learners reported that the dimension were not so well suited. The topics explored in this group are related to the capability of offering a suitable overview of the current status, to the absence of irrelevant information, to the level at which the visualisation can support the users reflection (also in comparison to the peers)

	Cod	Question	nr	MIN	MAX	Mean	SD
C1	USD1	It is easy to understand this visualisation	22	3	5	4.23	0.8125
	USD2 *	I find this visualisation unnecessarily complex	22	2	5	4.05	0.8439
	VBBI1	I think this visualisation would help instructors in tailoring their teaching to individual learners	21	2	5	3.90	0.8891
C2	UST1	I find this visualisation suitable for getting an overview on the current status in the learning process	22	2	5	3.86	0.7102
	UST2 *	I think the visualisation provides irrelevant information	22	2	5	3.86	1.0372
	VBM1	I think this visualisation can help learners to reflect on their learning	21	2	5	3.71	0.7838
	VBBP2	I think this visualisation can help learners to better understand their learning through comparison with other learners	21	1	5	3.71	1.0556
	VBLE1 *	I think the use of this visualisations will not make a difference on learning performance	21	1	5	3.62	1.0713
C3	VBM2 *	I think this visualisation will not significantly promote the learners' understanding and awareness of their learning progress	21	2	5	3.57	1.0282
	VBBI2 *	I don't think that this visualisation can help teachers in better understanding their students' needs	21	1	5	3.57	1.3628
C4	VBCL1	I think this visualisation is able to leverage mental workload	21	2	4	3.14	0.7270
	VBBP1 *	I think this visualisation would hinder collaboration among peers	21	2	5	3.10	1.1360
	VBCL2 *	I think interpreting this visualisation would put additional cognitive effort on the learner	21	1	5	3.00	0.8891

Table 5.1: The results of the survey about the OLM visualisation implemented in the experiment within the course "Educational Communication and eLearning": statistical analysis.

about their learning experience and the expected impact of the visualisation on their learning performances.

The following two groups share the fact that negative answers grow to a significant rate, positioning them between the (possibly or almost certainly) problematic aspects.

Group C3 comprises the questions named VBM2 and VBBI2, which are respectively related to the possibility of promoting learners understanding and awareness of their learning progress and to help instructors in better understanding the students' needs. Here the percentage of negative answers reached important levels.

C4, the most problematic set, is represented by the VBCL1, VBCL2, and VBBP1 questions which investigated the capability to leverage mental workload using GVIS as well as the additional cognitive effort imposed by the tool and the possibility that it may prevent collaboration amongst peers.

On the whole, the first test case demonstrated a good acceptance of the functionalities that were made available. There is however a need for more analysis and refinements in order to avoid side effects, especially in the areas covered by questions from groups C3 and C4.

5.1.2 An Experiment with Adapt2

The case with Adapt2 was a project developed in cooperation with the research team lead by prof. Peter Brusilovsky from the University of Pittsburgh. Throughout this work we explored the potential of presenting the students with a social visualisation of their performance. The visualisation tool that was developed as part of the project can be considered as an Open Social Student Model because it represents the student model alongside with a replica based on the class average. We compared the online learning behavior of two groups of students, on the same course and with the same instructor, over two different semesters. We showed that the students using visualisation were more engaged in the learning activities and achieved a better performance in self-assessment quizzes. We interpreted these facts under the light that the students are more conscious and serious when they are equipped with a visual representation of their performance. We also showed that the students in second group made more consistent efforts –measured as numbers of accesses to learning resources, but also to the self-evaluation quizzes– throughout the semester, and particularly so at the beginning of the semester.

5.1.2.1 The problem of engaging the students

According to the information provided in the website of Next Generation Learning Initiative [(http://www.nextgenlearning.org/the-program/)], only 42% of young people in the US who enroll in college complete a bachelors degree and just 12% of them complete an associate degree. On the other side, the panelist contacted to retrieve this data highlighted that, by the year 2018, 63% of all U.S. jobs will require some sort of postsecondary education. There is no reason to doubt that this will be the case for other developed countries too. This fact demonstrated the importance of engaging college students in learning. The present study showed that providing students with social performance visualisation could improve their engagement in learning and positively impact on their performance.

5.1.2.2 The Proposed Solution

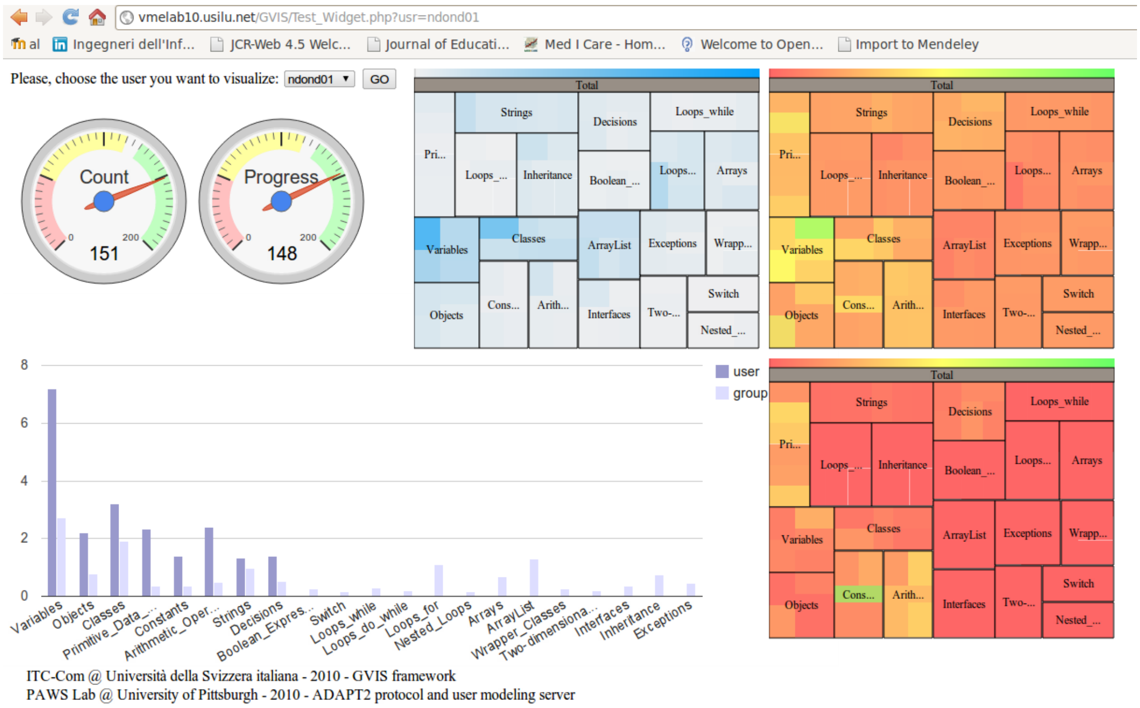

Figure 5.5: Initial mockup of the Student View for the implementation of KnowVis interface. The feedback received induced us to simplify and reduce the amount of information shown on the page and provide, instead, some navigational capabilities on the data.

Based on the theory of social comparison and social translucence (Erickson and Kellogg

2001) (Erickson, Halverson, Kellogg, Laff, and Wolf 2002) motivational visualisations can be used to encourage user participation in online communities. In this study the results of using a visualisation tool on students' engagement in online learning activities is presented. This visualisation tool exploited the actual web usage data of students in a learning support portal (KnowledgeTree) for C programming. KnowledgeTree (Brusilovsky 2004) is an adaptive repository of distributed learning resources that enables the instructors to present the learning material from different sources in a hierarchy of course objectives. These resources included lecture slides, program examples, instructor comments, self-assessment quizzes, etc. The portal carefully stores all the student activities and provides a fruitful resource for our student modeling engine. In this research the usage pattern of learning resources available in the learning portal between two groups of students was compared. The first group had access to a basic form of social navigation support (*control group*) while the other group was provided with an explicit form of social visualisation (in the form of Gauge, BarChart, TreeMap) to monitor their progress and compare themselves against the class average.

5.1.2.3 The Approach

To investigate the impact of social performance visualisation on students' performance, a social visualisation tool KnowVis was developed and deployed in the context of a C programming course. The visualisation tool was developed using a specialisation of the GVIS framework called KnowVis and it was made accessible to students through the course portal, KnowledgeTree (Brusilovsky 2004), which also provided access to a range of interactive learning resources delivered by activity servers QuizGuide (Brusilovsky, Sosnovsky, and Shcherbinina 2004), WebEx (Brusilovsky 2001), and NavEx (Yudelson 2001). QuizGuide provided the adaptive navigation support for self-assessment quizzes, while WebEx supported learning from annotated examples and NavEx provided the adaptive navigation for annotated examples.

KnowVis gathered student data by retrieving the student usage and performance logs maintained by the activity servers. After extracting the learning activities from such servers, KnowVis calculated the confirmed knowledge of each individual and of the group, based on the answers from the self-assessment quizzes.

To respect the principles of abstraction and progressive evaluation, the tool was structured on two levels. The first set of visualisations in KnowVis contained two gauges to indicate, respectively, the student's attempts and confirmed knowledge (Upper part of Fig. 5.6).

The variable called *Attempts* measured –through the number of tries on the self-evaluation quiz at the end of each elementary topic– the total number of accesses to learning objects performed by the student. On the other hand, the *confirmed knowledge* variable summarised the student's current knowledge as known by the ITS i.e. based on the answers provided in the self-assessment quizzes.

These two variables are then adjusted to the current class situation and divided by the class average. This way, these two succinct indexes not only represented the students' current performance on the course, but they also pointed out where they were standing in respect to the class as a whole. Each index represented on the gauge can be put through two other different visualisations, a BarChart (lower part of Fig. 5.6) and a TreeMap (lower part of Fig. 5.7). For example, once the BarChart Attempts is selected, each detailed learning activity will be presented by means of a bar representing the actual value. To give the students the opportunity to quickly locate their deficiencies and reflect on them, KnowVis also presented them with two side-by-side TreeMaps, one representing the performance of each individual student in the class and the other presenting the average performance of class. These detail views provided the opportunity for students to closely monitor their learning progress in a more fine-grained fashion and to effortlessly compare it against the class average.

5.1.2.4 Result: Basic Statistics

We expected that providing the students with social performance visualisation would increase their awareness of how they are doing in comparison to their peers and enable them to become more engaged in online learning activities. The main finding about this study was that the percentage of students that undertook self-assessment quizzes almost doubled, with an increase from 47% to 78%. We also found that the overall number of activities they performed on the system increased throughout the whole semester and that the students were more engaged while using the system. Table 5.2 shows the overall statistics on application usage within the system. The reason why we separated the QuizGuide from the other applications in the table is that there is an interesting pattern regarding the student's behavior on self-assessment quizzes (line *Average* number of attempts for the QuizGuide application). As we can see from the table, the average number of attempts to answer the self-assessment quizzes has significantly decreased for the students using visualisation i.e. from 335.28 to 186.93. In an effort to investigate this more deeply, we calculated the average time that the students spent on quizzes and their success rate. The overall statistics about the student performance in QuizGuide is summarised in Table 5.2.

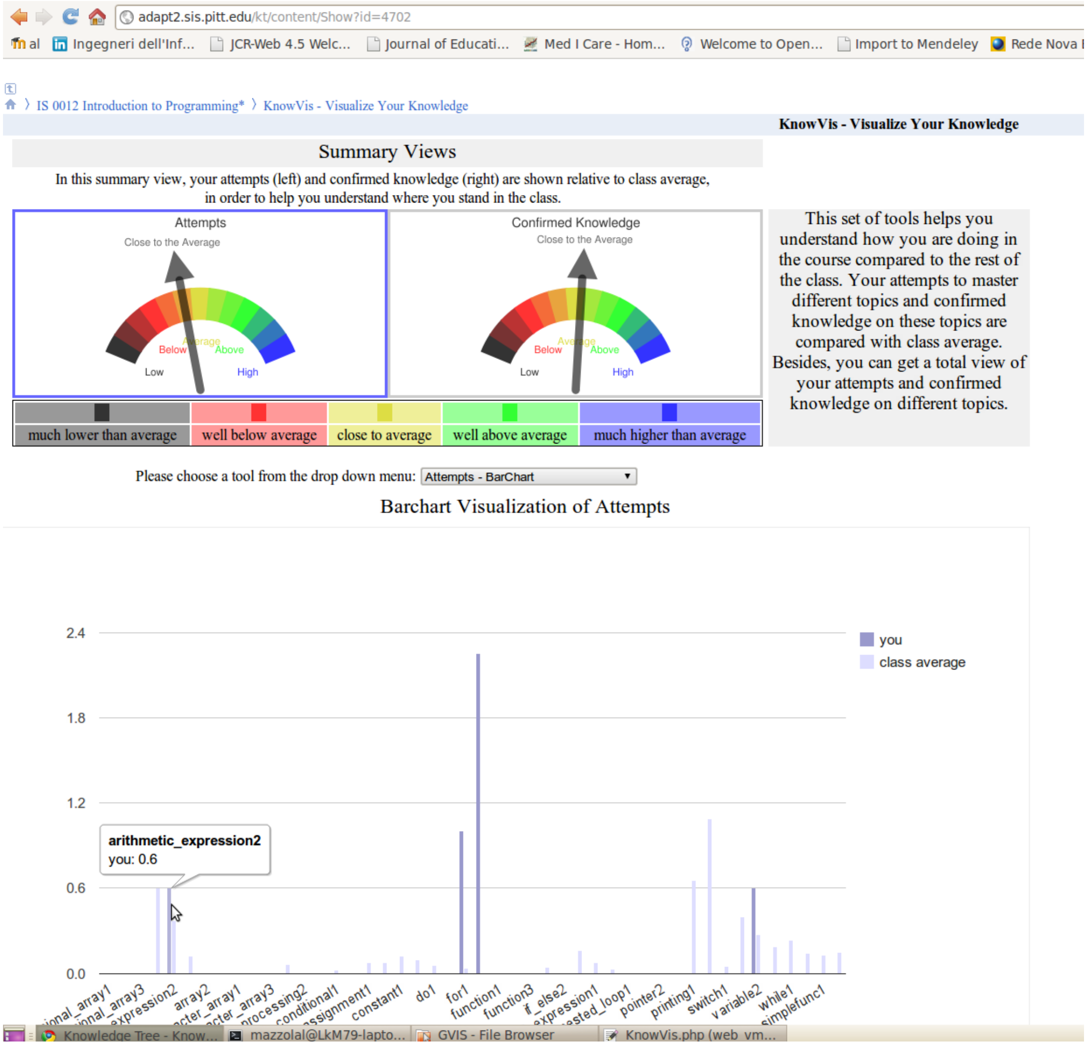

Figure 5.6: Student View of KnowVis.

The data showed that the average amount of time spent on self-assessment quizzes for each user increased by 20%. More importantly, the average success rate (number of correct answers divided by number of attempts) also increased by 9%.

We can consider these results as a sign that the students paid more attention and were more committed when they approached the system and spent more quality time solving the problems as they knew that it would directly affect their Progress visualisation. We can also assume that the social visualisation tool not only made them spend more time on the system but also created a sense of competition between them that resulted in a higher accuracy. Despite the fact that the above seems expression of an opposite feeling to the one presented in the previous case, it

		Without Visualisation	With Visualisation
		Fall 2010	**Spring 2011**
QuizGuide	#Users	7 out of 17 (41%)	15 out of 19 (78%)
	#Attempts	2347	2804
	Average	335.28	186.93
Other Apps	#Users	17	19
	#Records	9921	16081
	Average	583.59	846.37

Table 5.2: Overall statistics of application usage - part I

was the only way the students' behavior could be explained.

The results of the study demonstrated that the students are likely, as showed by the analysis of the log interaction with the system, to review their confirmed knowledge level as provided by the system and to perceive a sense of comparison between their progress and their peers'. This could be explained by the theory of social comparison.

We also investigated the number of sessions during which the students accessed the systems and the average time they spend in each session. The average number of sessions increased from 8.87 to 11.71 and average time spent by the students inside the system almost doubled, raising from 1679.84 seconds to 3314.41 seconds.

5.1.2.5 Result: Deeper Analysis

Although there were improvements in the system usage by the students, the basic statistic results did not show any significant improvement on the impact of KnowVis. Consequently, we tried to analyse the student data more deeply and to consider their usage pattern throughout the whole semester. In order to achieve this, we made a comparison between students of two course editions: Fall 2012 and Spring 2011.

The QuizGuide system –one of the course implementation done using the Adapt2 platform– was introduced to both groups of students at the beginning of the semester and both of them accessed to the system during the whole semester. The Fall 2010 group (without visualisation) accessed the system in a 94 days period and had one midterm exam on day 45. The Spring 2011 group (with visualisation) accessed the system in a 92 days period and had one midterm exam on day 44. We tried to investigate the daily behavior of the students during the semester. Each semester consists of 14 weeks.

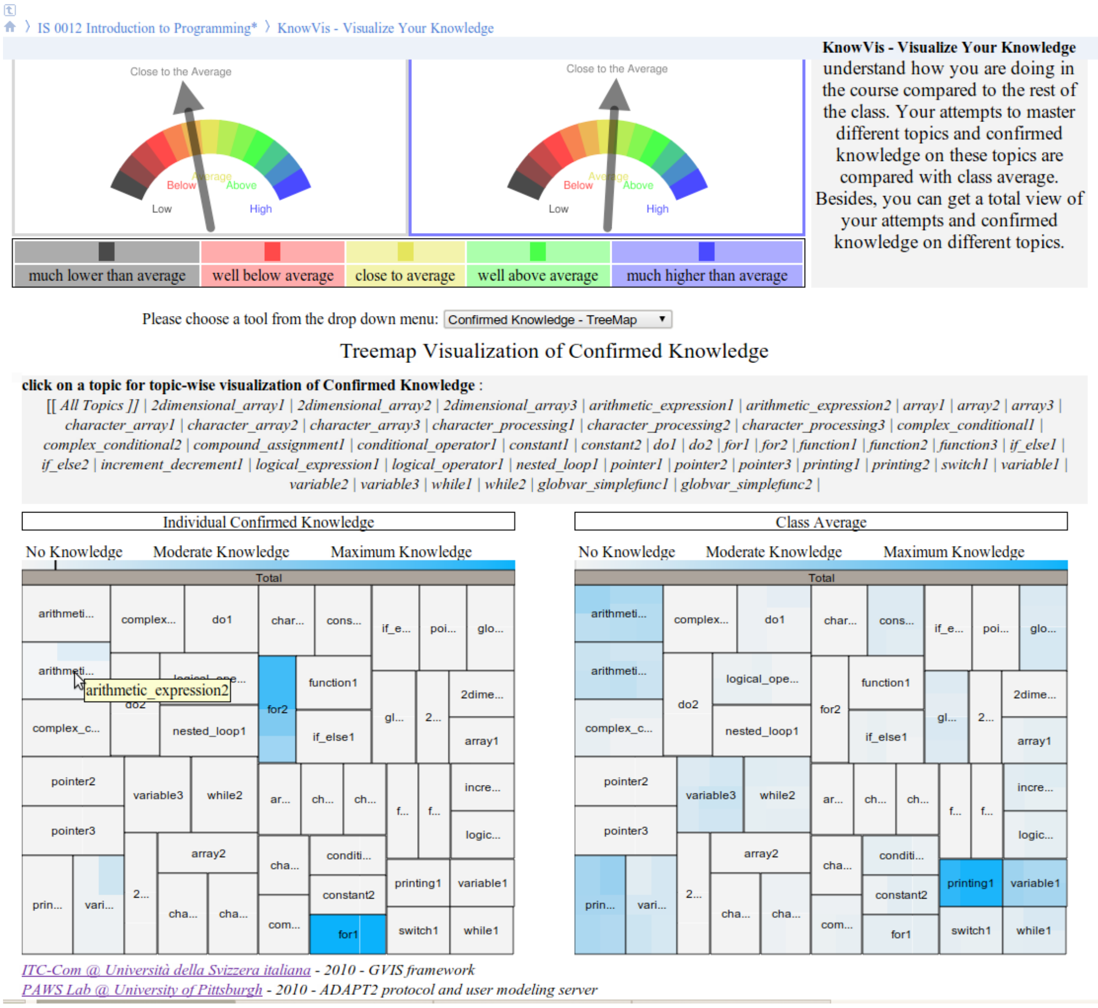

Figure 5.7: TreeMap Visualization of Attempts.

Fig. 5.8 shows the number of records for both groups as a weekly breakdown. Table 5.3 shows the results of the investigation. As it is shown by the data presented in the table, it's worth mentioning that the use of visualisation also improved the daily activities of students, even though not in a significant way. Looking at Fig. 5.8, we noticed that there is a sudden increase in the usage data from students in the group without visualisation (graph in the lower part) after midterm. This led us to looking at the students' usage data before the midterm and we found some significant increase before the midterm exam. Table 5.4 shows a brief summary of the results. As we can see from the table, the daily usage of the system within the first part of the semester (before midterm) significantly improved at all levels for the group that used visualisation.

This result could be interpreted as follows i.e. the early instalment of the social performance visualisation encourages consistent efforts –in term of system usage for studying purposes– particularly early on in the semester, which is really important because in that period of time the students have enough time to reflect on their weaknesses and keep up with their course load as efficiently as possible. As a result the students were better prepared during the whole semester, and in fact this hard work resulted in a higher success rate in answering the self-assessment quizzes.

Daily Usage Report	**Without Visualisation**	**With Visualisation**
	Fall 2010	**Spring 2011**
Sum. # Attempts	12280	18887
Avg. # Attempts	129.26	198.81
p value	$0.07 > 0.05$ (not significant)	
Sum. # Sessions	167	217
Avg. # Sessions	1.76	2.28
p value	$0.09 > 0.05$ (not significant)	
Avg. # Users	1.17	1.60
p value	$0.014 < 0.05$ (significant)	

Table 5.3: Daily usage record (whole semester)

Daily Usage Report	**Without Visualisation**	**With Visualisation**
	Fall 2010	**Spring 2011**
Sum. # Attempts	2091	18887
Avg. # Attempts	53.62	203.21
p value	$0.0006 < 0.05$ (significant)	
Sum. # Sessions	75	130
Avg. # Sessions	1.92	3.33
p value	$0.007 < 0.05$ (significant)	
Avg. # Users	1.23	2.03
p value	$0.0017 < 0.05$ (significant)	

Table 5.4: Daily usage record (before midterm)

5.1.2.6 Evaluation

To evaluate our visualisation tool, we conducted a thorough evaluation analysing the logs collected by the system adopted and the student results, of a semester-long classroom study. The

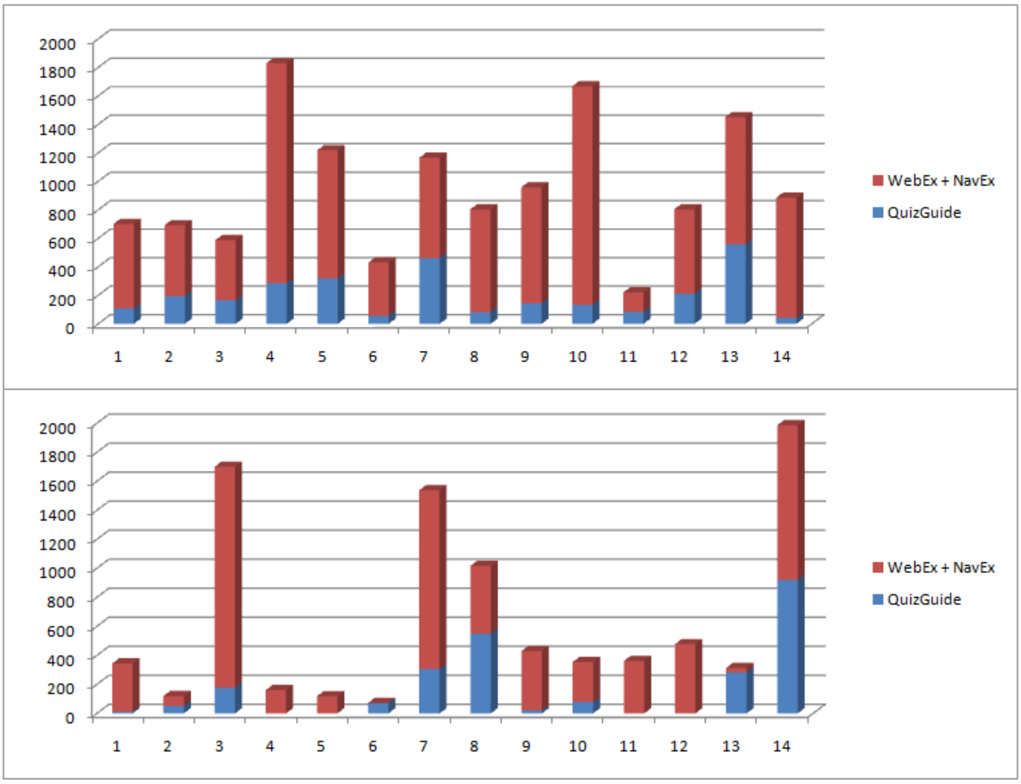

Figure 5.8: Weekly Distribution of Activities (Top with Visualization and Bottom without Visualization).

study was performed on an undergraduate Introduction to C Programming course, offered by the Community College of Allegheny County (PA), USA in the Spring Semester of 2011. To assess the impact of our visualisation tool, we compared the student usage data against a comparable class taught by the same instructor, with the same course structure and same personalised learning platform but without the visualisation tools, as previously stated. This group used the same learning resources bar the visualisation in the Fall Semester of 2010. All students had access to the same learning activity servers (QuizGuide, WebEx, and NavEx) through the KnowledgeTree course portal. All student activities within the system were recorded, including every student attempt to answer a question, read an example, study a line of the codes, etc. The system also stored a timestamp, the user's name, session ids, and the results of answers to the self-assessment quizzes (i.e. right or wrong).

To sum up, the visualisation provided presented a quite extensive usage: the number of sessions was around 150 and the tool provided was used by 30 distinct users along the whole

	Without Visualisation	**With Visualisation**
	Fall 2010	**Spring 2011**
Total # Attempts	2347	2804
Total # Questions	179	179
Average # Sessions	4.6	4.42
Average Time Spent	3013	2728.18
Average # Attempts	335.28	186.93
Average # Success	127	88.66
Average Success Rate	39%	48%
Std. Dev. Success Rate	0.089	0.13

Table 5.5: Overall statistics of application usage - Part II

semester. This was particularly interesting considering that the tool was not particularly marketed to students as a fundamental component across the extensive set of all the activities provided on the course throughout the semester, but rather it was just inserted –alongside the other course materials– in the reference page as one of the available resources for autonomous usage.

From the log analysis, three distinct general patterns stood out with regards to usage: some learners explored the capabilities of the tool at the very beginning, then monitored the evolution of their status with the compact indicator and, eventually, used the specialised treemaps and bargraphs to understand strength and issues of their profiles. Other learners just used the tool as a reference at the beginning and at the end of the subject study; they were not referring to it for controlling the evolution of their profile and they were using the other materials provided in the meantime. Finally, a number of learners used only the general compact indicators to gain an indication of the global status and their position in respect of the class average, and accessed it on a more or less regular basis.

When we consider all the learners involved in the study that accessed the tool at least once (i.e. 15 people), each one of them generated an average of 4.93 sessions (i.e. contiguous sets of actions, typically connected to a single study session), with an average page number access of 3.99 and accessing 1.71 of the capabilities provided (of the 5 globally available, that are the compact indicator and the 4 specific views: two for numbers of accesses and the others two for the confirmed knowledge) . The average number of pages visited per session is 1.92 with a length of each stay of around 71 seconds. These numbers demonstrate the learner perception of interest in the tool, that presents an average stay of 24 seconds for each page visited.

We believe that the lack of significant differences on the commonly acceptable level of $p < 0.05$ was caused by a relatively low number of students participating in the study.

5.2 Other application of GVIS outside LMS/ITS

Another test case that was carried out on the infrastructure did not involve any LMS or ITS, but aimed at practically demonstrating the possibility to use the GVIS tool for mashing up data from distributed and heterogeneous sources. For this example application we have mashed up URLs[1] from the user browsing history with tags coming from del.icio.us[2]. The resulting output, represented as a pie chart (see Figg. 5.14 and 5.15), shows the most relevant topics followed by a user. It can be consider as a sort of personal browsing profile. As an intermediate step we developed two simplified applications that rely on a subpart of the processing configuration. The first one builds a website profile starting from the tag associated to it, while the second presents the temporal evolution of these website profiles in order to evaluate the topics users perceive associated to it over time.

5.2.1 Ideas for another test case

A previous work from (Gwizdka and Bakelaar 2009) had already proposed to use a visualisation of the user navigation history in terms of interest areas based on a folksonomy. On this basis the present test case added a fully customisable approach for visualising it. As one would expect, some other approaches have already appeared in literature to represent the learner profile. Furthermore, the use of visual representations based on tag-cloud metaphors are quite common for presenting data based on keywords classification (Moulaison 2008).

More recently some attention has been devoted to the aspect of social interaction supported by online platforms, and as a result the representations provided have been modified accordingly (Mazzola and Mazza 2009a).

[1] Uniform Resource Locator, is the standard adopted to identify global resources also in Internet. It is also often called, improperly, the web address.

[2] del.icio.us is a social bookmarking service which allows logged in users to save resources identified by specific URLs by associating them to one or more *tags* as label to identify, classify and retrieve them at a later stage. It is sometimes referred to as Delicious, as will be the case throughout the rest of the chapter. The project has now been closed but some dumps are available for study and research purposes.

5.2.2 The approach

For this experiment part of users profiles was disclosed to the users themselves, showing information related to their browsing habits. It was decided to work on URLs as they represent a type of data which is very easy to mash-up between different systems: every URL points to one specific Web resource, and there is already plenty of metadata available about them on the Internet.

Creating mash-up required, first of all, to identify the main data sources. In order to analyse users browsing habits a minimum of two different sources were required: a collection of logs capturing the visited URLs and some kind of classification of the matching Web pages.

Within an E-Learning system (where user clicks are normally collected by the system for tracking and debug purposes) it may not be so difficult to obtain both the data sources. Assuming that the pages used inside the platform are stable and local, the classification –given the limited size of the system (associable to a closed corpus of documents)– could be built top-down, as a taxonomy.

However, the task could became more complex when captured URLs represent generic pages on the Web, such as external links provided within the system itself (associable to a open corpus of documents): in this case a top-down approach is not suitable, while a bottom-up categorisation, such as the one provided by a folksonomy, might offer better results.

For this particular project it was decided to tackle the problem of generic URLs, for a number of reasons. First of all, most of the public Web pages (that we define "in the wild" Internet), as opposed to the ones within the E-Learning system, can be assumed to be accessible from anywhere. This is mostly untrue for the other case, especially if the system provides restricted access. Even if metadata is available for pages in a closed environment (they are still represented by valid URLs, so annotations are possible), there may just not be enough of it to make it statistically relevant. Finally, datasets for generic URLs are easily available as they are automatically saved by most browsers in the "browsing history". To make the work easier, it was chosen to use just one browser and to collect Firefox history database for the tests. With a generic browsing history as data source, to use folksonomies as a source of metadata seemed the most suitable choice. In these systems (named after folks and taxonomies) users can associate freely chosen tags to web resources, producing knowledge which is useful for them but also made available to the entire community (Quintarelli 2005).

As the work of categorisation is performed by the users themselves, folksonomies are democratic, scalable, current, inclusive and have a very low cost (Kroski 2005).

Of course, this bottom-up process has some drawbacks too: due to the absence of a unique, coherent point of view, tags cannot be easily organised in a hierarchy. Also there is no synonym control and systems might lack both in precision and recall (Halpin, Robu, and Shepard 2006).

Aware of both advantages and limits of folksonomies, it was decided to adopt Delicious as the tag provider due to the facts that it already has a huge quantity of metadata and is able to provide results which are statistically good.

5.2.3 System Infrastructure

The output of the tool as seen by the final user is a flash object that represents, with a pie chart metaphor, the relative frequency of every tag. The extractor configuration for the database source only requires three different kinds of parameters: the authentication credentials, a SQL query and the format desidered for the output, which is piped to the aggregator module. Similarly, the extractor configuration for the SPARQL source, used to access the folksonomy, only requires the endpoint access information, a SPARQL query and the output format.

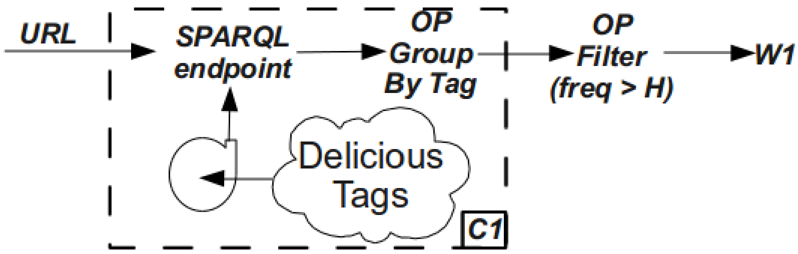

Figure 5.9: The operation pipelines applied to the actual examples: C1 is the part which is common to all the applications developed for this test case.

Delicious did not provide access to its tags through a SPARQL endpoint, rather it exposed an API[1] which, given an URL, returns a list of the latest tags that have been used to categorise it. The information returned by the API was structured and serialised in a standard format (JSON), however it did not share its schema with other tag-based systems, nor there were any plans to implement this in the future. As a result, there would be the need to write a new adapter for GVIS every time it was necessary to access tags using another Web service.

However, what we wanted was to be able to change, at a later stage, the source of the folksonomy without having to write a new GVIS class adapter for the specific API offered by

[1]Application Program Interface is the signature of the methods that can be called by any external system that wants to communicate with it. It exposes the methods and the parameters required to obtain a certain function.

the newly chosen source. To address this problem, a more generic conversion tool was built which took Delicious API results as an input, converted them into RDF on the fly (following a given ontology schema), and finally exposed them as a SPARQL endpoint (using Joseki, http://www.joseki.org). The advantages of this tool were manifold: first of all, it allowed GVIS to access tag information from a generic SPARQL endpoint, using the same query regardless of which folksonomy was being queried. As a result, information was not only independent from the data source but it was also easier to merge inside one single place, thus providing something that was not previously available (the union of different tag-based systems). Finally, as its code was not bound to this particular application, it could have been reused for other purposes (as it was actually done while developing the prototype for a semantic annotation system). Furthermore, with the development of a SPARQL adapter GVIS became capable of replacing the existing folksonomy data-source (Delicious) with any other one that conforms to SPARQL by simply allowing for the query to take the new source semantic into account.

5.2.3.1 Data Aggregation

GVIS provided a set of operators to support data aggregation "out of the box". These operators implemented some common aggregation patterns such as group by, filter by threshold, order by, etc.

Once the raw data was available, the following step consisted of applying a subset of operators in charge of expressing the transformation logic from source data to final information. The sequence of operations had a basic common part (named C1, see Figure 5.9), used to retrieve del.icio.us tags from the SPARQL endpoint as well as a part which was specific to the browsing history, shown as C2 in Fig. 5.10.

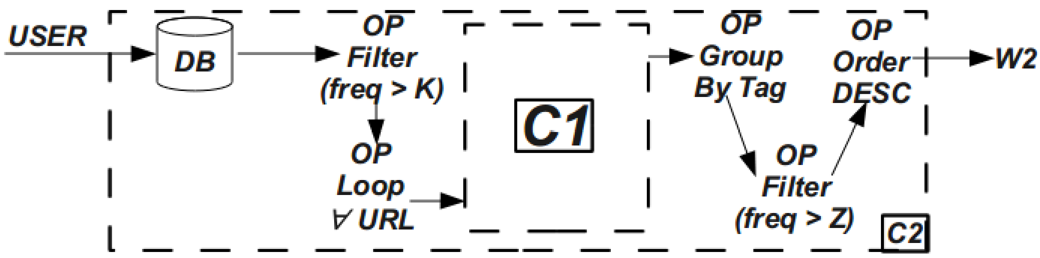

Figure 5.10: The operation pipelines applied to the first examples: C2 represents the logical operations to produce the browser history navigation chart (the starting point is the browsing history provided by the user; here the USER in input represents his identification).

C1 only contained a grouping operator, which allowed tags to be grouped and counted when information about a single URL was requested. C2 was more complex and it was used to categorise the whole browsing history: it first used a filter operator, following the extraction of visited domains from the DB, to filter the most visited ones. It subsequently applied a loop operator to get tags related to every single URL in the list and, after completing the aggregation of common entries (tags), it applied another threshold to remove the less important tags (ie: the ones with low numerousness). Finally, an order by operator was used to return the tags ordered by occurrences (needed by the visualisation component to show the pie slices in decreasing order of magnitude).

5.2.3.2 Visualization

The visualisation was implemented using a pie chart metaphor, as shown in Fig. 5.11.

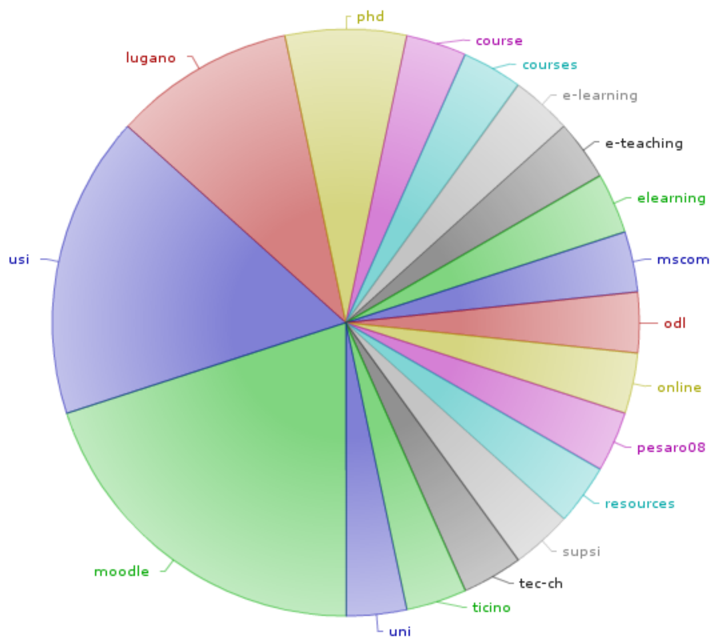

Figure 5.11: An analysis on a website: the case of http://corsi.elearninglab.org (the eLearning platform of the university of Lugano).

This choice represented one of the possible visualisations available within the system, as provided by an open source library called Open Flash Chart. Of course, thanks to the modularity of the tool, it would have been possible to import other libraries to provide more common metaphors for tags such as tag clouds or weighted lists.

However, for this particular case the pie chart seemed to be the most suitable metaphor, as it does not just show magnitudes but also indicates a relative weight between different interests in the user profile. Moreover, the particular widget chosen offers the possibility to interactively explore the detail of every single result by presenting more data on user interaction, such as the number of times the specific tag was counted.

5.2.4 Some examples

Exploiting the operands described in the data aggregation section we built two different types of applications. The first one was simpler, accepting a single URL as input and returning a pie chart depicting its most frequent tags as output. The second one was more complex: its input was the whole browsing history of a specific user and, exploiting the component previously described as C2, it returned a pie chart depicting the user's main areas of interest.

5.2.4.1 Websites

The first application, that can be used to have a quick glance at the main tags characterising a website, was very simple and comparable, in terms of information provided, with the original delicious web interface.

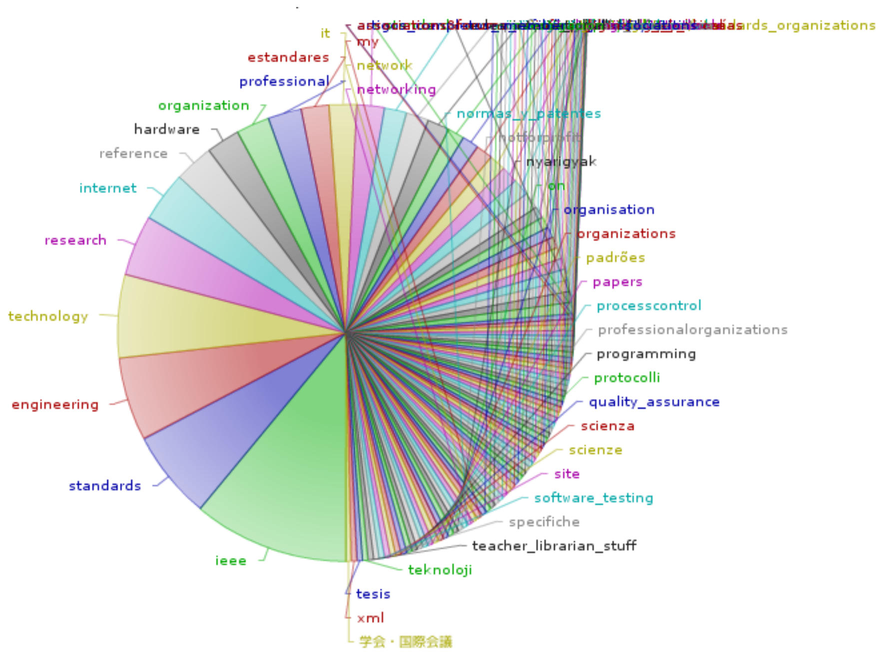

Figure 5.12: An analysis on a website: the case of http://www.ieee.org (the IEEE organisation institutional website): the issue of the overabundance of data to represent is clearly evident.

This was however enriched by a graphical presentation of the data, an alternative to the classical tag cloud: at a first glance, it gave an immediate idea of what the most important keywords were, along with their related weights; then, moving the mouse on a pie slice, it was possible to see how many tags had been returned by the system and which percentage of the whole set they covered. In Figures 5.11 and 5.12 two different websites are compared. The figure 5.11, with a relatively low number of keywords, is characterised by a very effective visualisation.

Conversely, in figure 5.12 an unfiltered view of a more notorious portal is depicted, showing how the number of tags returned by the system can negatively affect visualisation, both from a technical perspective (the library was not capable of dealing with all that data) and also from the users point of view (information was unreadable or too much dense to be correctly interpreted).

This result justified the use made of threshold levels in order to simplify the output i.e. trying to dig out useful information emerge from a huge amount of data. Another experiment made with single URLs was around transient behaviors: the evolution of a newly created website for a conference was monitored and followed in its growth, at intervals of one month's time, using the tool.

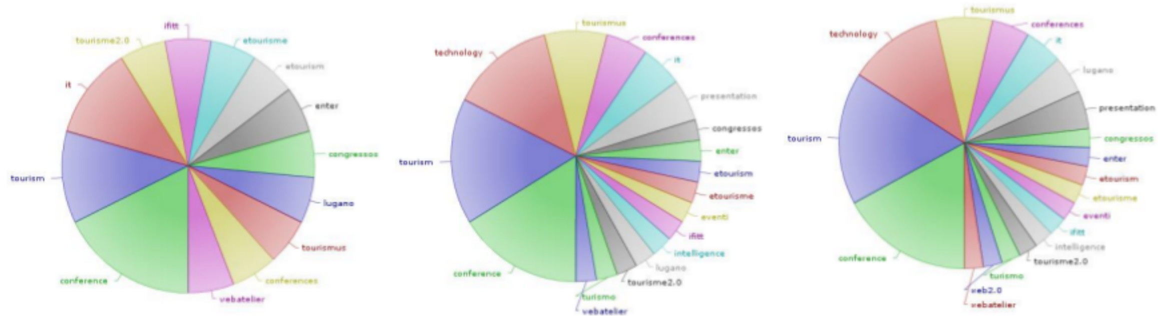

Figure 5.13: The ENTER2010 conference website (http://www.enter2010.org) analysed in three different moments: (from left to right) an early one, just after the conference announcement, after a month and two months from creation.

Figure 5.13 shows three screen-shots taken at three different times: the first one corresponds to an early stage of life of the websites, the middle one was taken after a month, and the last one after two months. The three images show an interesting result: as time passed, the most important subjects of the conference emerged. However, in the long run the relative weights of the main areas seemed to remain stable.

5.2.4.2 Browsing history

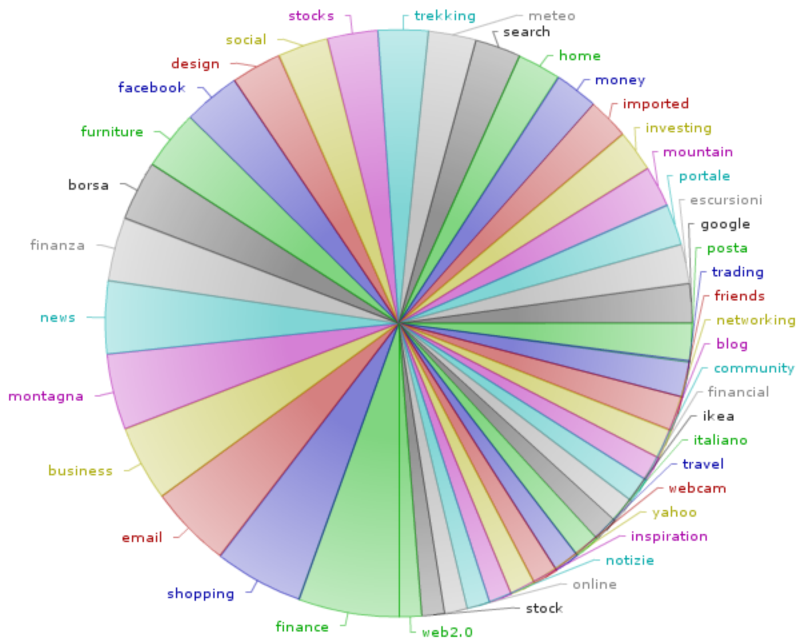

Figure 5.14: The browsing history for one user: a person with interest in economy and trekking (later recognized to be associated with an adult investing and that likes mountains).

The second application developed showed the most relevant topics, in terms of tag frequency, characterising some users browsing habits. While the visualisation appears identical to the previous one, the process required to build it was much more complex. To test the system with real data, various users were asked to supply a copy of their Firefox history files.

To protect personal information, the user data was anonymised by mapping potentially sensible URLs (containing user-dependent parameters) to their matching domains and the final result was built from this anonymised data. The anonymised data was then connected with the tags associated to it in the folksonomy chosen and their number was counted.

Figures 5.14 and 5.15 show two profiles in which the most interesting topics for each users were quite evident (in the first one - Fig. 5.14 - finance and mountain/trekking, while in the second one - Fig 5.15 - video streaming resources). Through the following phase –where the created profile was shown to the user it belongs to– we also had the chance to learn that the first one is an adult while the second one is a younger person.

As one would expect, other less personal tags were also present such as google, search, email, and web2.0 (meaning that these users were not actually interested in these topics per se but

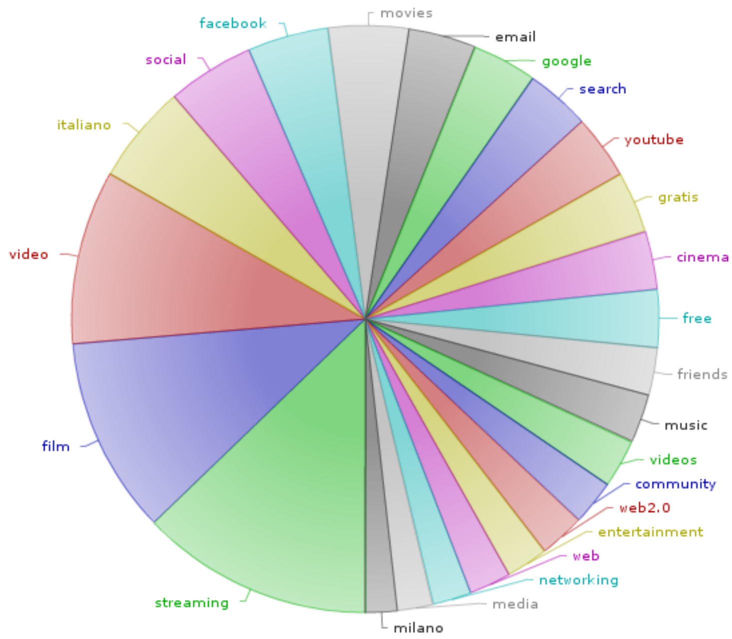

Figure 5.15: The browsing history of another user: an individual mainly watching movies on streaming (probably a young person, as it was later confirmed through user self-disclosure).

rather they used the related search engines, email web clients and social websites).

In another sub–experiment, the charts obtained from two "high tech" profiles (i.e. people working with ICT to provide advanced services for university institutions) were compared. The peculiarity of this experiment was that both users had a huge history file and the resources they browsed were normally bookmarked by many people. As a result, the total number of tags collected was much higher than in previous examples.

In spite of the similarities between the two users (and despite the filters we had to apply to avoid ending up with too many tags to visualise) it was still possible to identify profiles specificities: in one widget an interest in open source emerged (with keywords –extracted from a categorisation of the delicious tags– like "opensource", "ubuntu", "linux", "free", "distros", and so on) together with software development in the form of scripting languages (tags: "shell", "programming", "scripting", "zsh"), while in the other widget a less specific profile emerged which was characterised by a higher number of tags related to areas like "Linux", "Moodle" and "XML", but with no predominant one.

5.2.4.3 The "reverse profile" of a domain name

Another application that was developed took into account the outlinks from a specific website to create a profile of its most relevant areas of interest: it worked like a sort of *visual pagerank*. The idea behind this experiment was that, for some categories of websites such as blogs, outlinks might express interest in some specific URLs from the user. In less personal websites, instead, outlinks could still provide information about the main topics they deal with.

The most interesting aspect was that, in both cases, data was freely available on the Web. Outlinks can be collected in many different ways, i.e. by running crawlers that visit single Web pages or traverse full websites. To quickly build a prototype we relied on an existing service i.e. the one provided by Bing with the *"linkfromdomain"* search parameter, which easily allows to retrieve all the outlinks from a given domain. A limitation of this operator is that it only accepts first-level domains, meaning that it was not possible to use websites with more complex addresses or subdomains branches as search parameter. However, this intrinsic limitation imposed by the tool wasn't much of an impediment as the most interesting cases with respect to this analysis were represented by first-level domains.

In order to create the application, a new operational pipeline C3 was developed (see Figure 5.16) which was quite similar to C2, except for one characteristic: the initial seed was a domain address that was passed to an extractor module specifically designed to call the search engine, perform the out-links search and parse its results.

Even though the theoretical number of websites the tool can be tested against was reduced by the aforementioned limitation of Bing, and despite the fact that the validity of such an approach has yet to be fully demonstrated, some initial tests showed that this approach offers both an alternative and complementary view to the one described in the second experiment. While the former experiment showed an explicit classifications made by the users about a website, in the latter it was possible to obtain an implicit description of a site according to the classification offered by the tags of other pages it was linked to. The process worked in the very same way by collecting, adding up and filtering by numerousness the tags associated to the out-links found, to ultimately create the final topics profile.

5.2.5 System Evaluation

As the purpose of this case study was to both show the methodology for the creation of graphical mashups and to describe its application to a specific case, we divided the evaluation in two

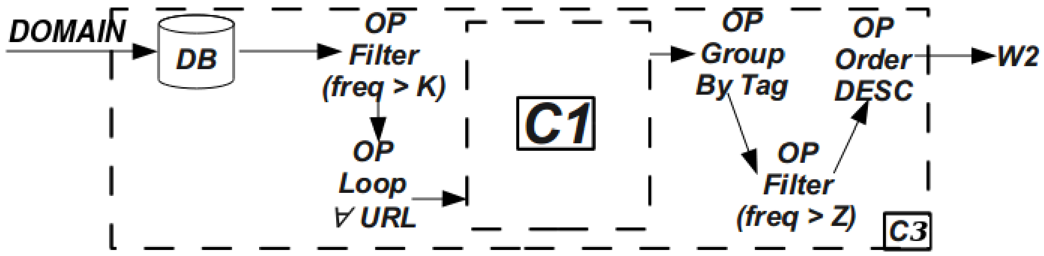

Figure 5.16: The operations pipeline applied to the actual examples: C3 encodes logical operations to produce the domain-related topics association chart.

different parts: the first one which mainly deals with the performance, stability and ease of implementation of the system, and the second one which is primarily concerned with the user's impression on the graphical representations obtained. With regards to the system, the new tool has been created with less than 180 lines of XML code, of which about 60 belonged to the extraction module, 90 to aggregation and 30 to visualisation. The fact that the aggregation part is bigger does not come as a surprise as it implements the pipeline of operators that group and filter URLs and tags. Of course the Delicious conversion tool has to be taken into account too, but it is worth noticing that the only operation it performed was to convert information from a structured format to another (albeit more expressive) one. As an added value, by applying this additional step we gave the system the capability to merge tags belonging to different folksonomies. The system was stable and reactive enough to create any profile in a time-frame of about five minutes in almost all the cases.

5.2.5.1 System Performances

System performance were also meeting our expectations, especially considering that no caching mechanism was implemented, which means that retrieving metadata for each URL required a new independent connection to the Delicious API.

Table 5.6 shows information related to the creation of user profiles representations. The table columns detail, respectively: the user ID, the number of domains considered (i.e. must have been accessed at least 10 times), the average number of del.icio.us tags for each selected domain and the average time required to collect all the data over ten successive executions. We only focused on download time as it accounted for most of the processing time (experimentally estimated as being more than 90% of the total execution time on a normal office workstation on a corporate/university Internet connection).

User	Domains	Tag/Domain	Time (sec)	Sec/Domain
01	70	52	211	3.01
02	125	37	200	1.60
03	223	39	385	1.73
04	23	33	32	1.39
05	49	20	60	1.22
06	84	34	130	1.55
07	89	38	125	1.40
08	11	72	27	2.45
09	12	42	17	1.42
10	72	37	116	1.62

Table 5.6: Performance Evaluation of the Tag-retrieval component. It is solely responsible for the most part of the total execution time.

As shown by the last column of the data table, there was a positive correlation between the number of selected domains and the total download time (as each domain required a Web connection). This was also influenced by the number of tags that matched every single domain (as the downloaded data increased with tags). Of course, there were other factors that this analysis did not take into account, such as the quality of the Internet connection, the load on the accessed servers and so on. In conclusion it can be said that the overhead introduced by the real-time access to external sources did not significantly affect the visualisation of a individual website, whereas showing a large user history might required up to several minutes.

5.2.5.2 Visualisations

To test the effectiveness of the visualisation a heterogeneous group of users was first created which was composed of ten people of different ages and with diverse interests and they were asked to supply their Firefox history for analysis. The panel was composed of students, technicians, and researcher at the eLab –the laboratory for e-learning applications of the University of Lugano– and it was recruited in the summer semester of the 2009 academic year. After each single profile was processed, the resulting pie chart was shown to each one of them and they were prompted to provide feedback on its accuracy and representativeness as well as on the interest/usefulness as perceived by the single user. The result was that all of them recognised their most relevant areas of interest on the pie chart, but not all of them showed full confidence in the results. Some were surprised about the presence in their profile of some specific tag keywords while others indicated, even though less frequently, the absence of one that was expected.

Almost all of the panel members demonstrated a vivid interest in the graph and indicated that this kind of visualisation could be useful to have as it would allow people to reflect on their interests and navigation habits, if purged from the noise generated by the less personal/more general tags.

A deeper analysis was performed on the most technically skilled individuals. Besides confirming the validity of the generated profile, they told us they were already aware that most of the tags presented can be associated to their profile. Nevertheless they expressed interest for the full global "picture" offered by their navigational habits.

Despite the positive aspects, there were still a couple of issues that had to be addressed to make the tool actually useful. Given the demonstrative purpose of the experiments, we decided not to proceed to solve these issues but rather we just to take the feedback provided onboard for reflection purposes. The first item of feedback received has to do with visualisation: as shown in Fig. 5.12, pie charts suffer information overload and become very difficult to read when the number of different tags characterizing either a website or a browsing history becomes too high.

A number of solutions exist for this problem, which either deal with the widget itself or with the filters applied to the data that is being visualised. With regards to the widget the adaptive configuration feature provided by GVIS could be applied: depending on the quantity of information available, different widgets could be used to always show results in the best possible way. The only limitation is that the widget library does not provide valid alternatives to pie charts, which would be required in case of huge quantities of data. Based on this limitation it was decided to apply a filtering solution, hiding the tags that were less used i.e. below pre-defined thresholds, which are deemed as less relevant in the profile composition.

Another interesting experiment was devoted to understand how to use the visualisation produced to explore the space of the folksonomies that the crowd would associate to a website. An example of the above is presented in Figure 5.17, where some of the synonyms, languages translations and variations in the tags set associated by the user (in Delicious) to the analysed website are evident (i.e: {italy, italia, italian, italia.it}; {portale, portale_turistico, tourism, turisti}; {sito, sito_web, web}; {informazioni, reference, official}; etc.). Here another possible improvement emerged i.e. the possibility to merge these sets into a representative unique tag –working as the *centroid* of the cluster– and to use this new vocabulary to create the profile, thus relying on a more abstract sets of topics.

Some in-depth analyses we carried out in this field can be found in the following publications (Dattolo, Eynard and Mazzola 2011) and (Eynard, Mazzola, and Dattolo 2012).

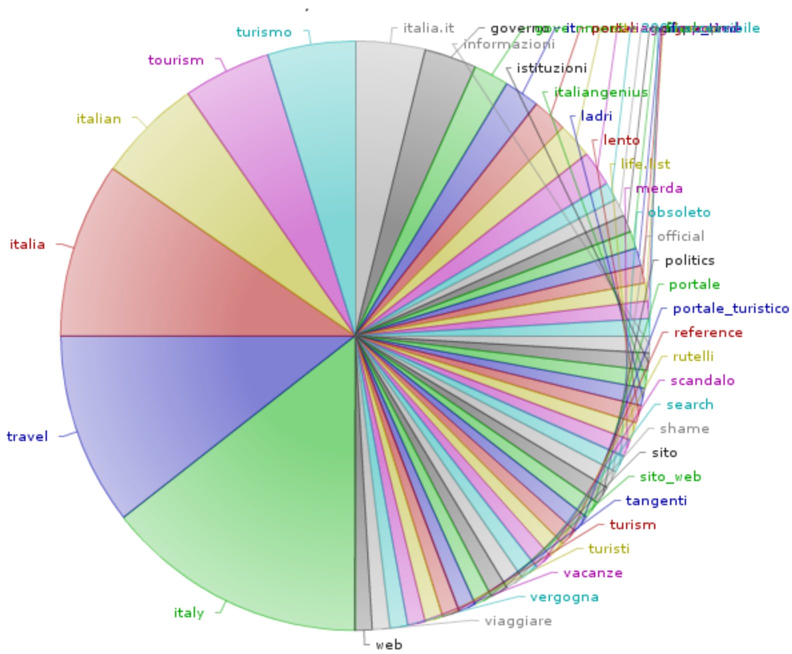

Figure 5.17: The www.italia.org website (http://www.italia.org) analysed in respect of the most frequent labels associated in the del.icio.us service by the users: the synonims, languages translations or variations present in the tags set are evident

5.2.6 Comments

This section presented some application examples related to the possibility to open up profiles, focusing on a specific aspect of user profile represented by the browsing history. A synthetic view of users main interests was built through the display of a pie chart visualising the main tags associated with the most visited URLs. A set of users who have tested the system with their own browsing histories reported that they were able to recognise themselves in the profiles that were presented to them. Some final remarks were focused towards two different directions: one related to the GVIS system as a whole and how it performed in this particular application, and another one which is more related to the application itself.

From the system perspective, GVIS performed well as an aggregator of heterogeneous sources, allowing to build the two examples previously shown simply by specifying a few lines of XML. The system was versatile enough to access different families of data sources (such as SQL databases, SPARQL endpoints and web services) and expressive enough to provide the operators needed to group, order, filter and mash up the results coming from the two sources queried. On the other hand, the long timeframes required to collect tags from the users' browsing history databases clearly demonstrated the need for a caching system or a tool to answer parallel queries.

With regards to the specific application developed with GVIS, the results seemed encouraging and worth further introspection. In particular, future efforts could be devoted towards the following directions:

- information visualisation – learn how to better exploit the information retrieved and computed in order to provide a compact yet expressive graphical representation of a user profile
- tag semantics – understand how to use it to cluster tags together and into families that can better describe the users' interests
- an in-depth user evaluation – rely on structured interviews, which should not only provide information about the correctness of the profile widgets but also on its usefulness in the context of the specific system they are embedded into.

Chapter 6

Conclusion

6.1 Discussion

The main aim of this work was to investigate how the introduction of an adaptive visualisation tool for Open Social Learner Models is perceived by the user and how it could impact on the learner's experience within an educational context. In order to achieve this objective, a tool called GVIS was designed and developed and a number of test cases – both on possible aspects/capabilities and on the implemented functionalities – were designed and executed. By relying on the data collected during the experiments it was possible to get an initial idea about the impact that the introduction of such a tool would bring.

First of all, the application of the developed tool demonstrated to have an impact on the behavior of online learners when used to provide them with indicators around their activities (for self-reflection purpose) [see section 4.2.3.1 for results on mock-ups evaluation and 5.1.1.2 for an implementation experiment results], especially when enhanced with social capabilities, such as the one to show the learner's status with respect of the rest of the class/group [see 5.1.2.4, 5.1.2.5, and 5.1.2.6 for the experiment results].

The effects appear to be amplified in those cases where the widget usage is as simplified as possible, with the possibility to be expanded and explored by the single user on an autonomous basis [see paragraph 4.2.3 on user feedback on mock-ups]. One of the possible drawbacks we identified is that cognitive overload can be generated on the new learners by this adjunct information. Despite the above, the data collected with post usage questionnaires shows that users generally consider this kind of support useful, as shown on paragraph 4.2.3.

On the tutor/teacher side there is an open issue around analysing impact and consequences to provide instructors with some sort of monitoring board. An initial attempt was made, although informally, in the experiment described in section 5.1.1. Even though the data collected in such a fashion does not allow us to draw any conclusions, it still confirms that the direction taken is promising, at least with regards to giving them a sense of what is happening on the online course in a prompt and immediate way.

All in all, this kind of approach demonstrated to be useful to create indicators [see sections 5.1.1, 5.1.2.2, and 5.2.6], even though the data collected seems to support this conclusion more for learners than for tutors/teachers. The first piece of evidence supported the idea –in adherence with what suggested by the mock-up feedbacks– that simpler and immediate widgets showed a higher impact, as presented by the experiments carried out with Moodle [paragraph 5.2.1] and Adapt2 [paragraph 5.2.2].

On the teacher/tutor side, even though the amount of feedback and usage data collected was lower than expected but we can still provide an initial idea about the effort required and the issues faced in the course preparation/adaptation activity to make it suitable for inclusion of the GVIS tool. The initial experience seems to indicate a positive perception, regardless of the difficulties connected with the need to elicit the semantic description of the data (i.e. XML configuration files) extracted or received from the environment. In an effort to avoid this issue, we wrote a number configurations for different systems, thus demonstrating the flexibility and adaptability of the tool in respect of the system it is connected with (LMS or ITS) too.

In one of the experiment performed (see paragraph 5.1.2.5), the data collected through self-evaluation questionnaires and pre-post test confrontations seem to also indicate a possible positive impact on the concept acquisition on the autonomous-learner side. Possible drawbacks that could emerge need to be explored such as the gaming of the system[1] or the social engineering[2] based on the social data disclosed.

The results obtained can be summarised as follows:

- A service that integrates and represents data – extracted from eLearning platform logs– seem to be potentially useful for an online educational environment as it would empower

[1]it is defined as *system gaming* when a learner tries to understand and make a better use of the internal mechanism of the system to obtain better result in the evaluations or quizzes, without acquiring more knowledge on the topic but rather just by knowing the rules that regulate the automatic evaluations carried out by the system, i.e. for a quiz evaluated through the points collected in the most recent tentative, trying the tests several times and memorise the correct answer to each question.

[2]it is defined as *social engineering* the secondary usage of social information to gain some sort of advantage, such as following the most performing learner and supporting/reusing the information they provide in a forum to build a cumulative paper around the topic, or showing friendliness to obtain support on the assignments to be delivered in the course.

it with social motivation and other aspects.

- An XML-based approach of "semantic" description about sources, data, operation and graphical mapping is in line with the objective of empowering the Instructional Designer with a useful way to interact and design the widgets to implement smart indicators of one or more user activities, with the possibility to check the formal validity of the configuration files produced through the usage of the DTDs presented at the end of this work, in Appendix A. Furthermore, this approach provides a set of predefined and shared configurations for different didactic models on well known and widespread educational infrastructures which allows to reduce the barriers to entry when it comes to start using the system. Despite these efforts, we are aware that a piece is lost to make the solution really useful: a graphical interface that supports two main functions. Firstly, it will allows the ID to define and check the configurations without the need of writing an XML piece by hand, where on the other side it can validate step-by-step at run-time the generated configuration using the provided DTD.

- Learners seem to appreciate the presence of personal visual indicator of their experience both in their perception of usefulness and in the commitment demonstrated towards the online experience, resulting in a moderate impact on the learning gain as showed by the self-evaluation test results in pre and post experience scenarios.

- The statistics in paragraph 5.2.1.5 (even though they are just initial results) suggest that the presence of these "smart indicators" (in the way they are defined in the work of Glahn 2009) could work as a social pressure tool by means of data collection and representation of the learner's actual status in the educational environment, for example when compared with their peers. This effect could be stressed and used to engage the learner since the earliest phase of the online educational experiences.

- Introducing adaptivity in the smart indicator can help reduce the cognitive overload generated by this new data source made available to the user and can enable them to fully enjoy the informative richness that can only be offered "on request" as they become familiar with the tool.

- The presence of this kind of visualisations assumes a higher relevance when we move towards a life-long PLE, working as a possible direction and feedback channel for learners and supporting them as a self-awareness tool. At the same time, they can act as a social mirror - in the way considered in the following IBM researches, and others (Erickson and Kellogg 2000), (Erickson, Halverson, Kellogg, Laff, and Wolf 2002), and (Bongwon, Chi, Kittu, and Pendleton 2008)) i.e. providing controlled information about the status of the class and the relative positioning of the learner in respect of his peers.

Some of the issues that lead to the development of GVIS were recently covered by a new standard for learning experience tracking called *TinCanAPI*, whose specifications are publicly available on the website `http://tincanapi.com`. It is an evolution of the SCORM standard[1] and it was first publicly released in mid 2013 (taking the name of *xAPI*).

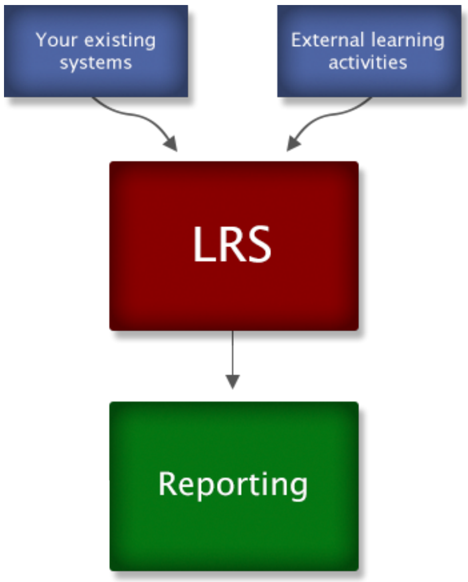

Figure 6.1: The xAPI (former TinCanAPI) working schema: all the *Activity streams* flows to the *Learning Record Store* (LRS) that then can be used to feed any reporting system.

An image that represent the behaviour that xAPI can offer is represented in Fig. 6.1. This solution addresses some of the issue presented in the current work, as depicted in the above graph. The behaviour is as follows – multiple sources of *Activity streams* sets are generated, in the form of <*subject - verb - object*>, independently from the server. The flows are then communicated to one or more *Learning Record Store* (LRS) for storage purposes. The same LRSes can then be queried to retrieve all the information previously archived, which can be used to feed any reporting system. Despite the advantages mentioned, and while GVIS maintains its

[1]Sharable Content Object Reference Model is a set of reference specifications for web based e-learning materials. It specifies the communication between the learning material and the hosting platform but also the format for the portable packaging of the learning object.

validity and usability in the domain, the current standard does not address the aspects related to adaptivity and graphical representation of information.

6.2 Recap

Throughout the present work an infrastructure was presented for the aggregation of information coming from different sources to create adaptive graphical presentations aimed at supporting the teaching and learning needs in technology enhanced learning environments. In the context of Life Long Learning, the presentation of this contextual information to the learners and to the teachers or tutors assumes a significant importance as it allows for a greater awareness of the learning situation and helps promoting participation. The possibility to include adaptation in the generation of the widgets could help avoiding the informative overloading, which is a critical aspect of the learning context. The encoding of this information in a graphical format is also important in order to make it useful and usable during the learning process. The architecture described seems to be useful to enable instructional designers to create graphical representations of one or more characteristics of the learner model. The semantic is expressed through a number of configuration files that drive the behavior of the component. We collected feedbacks from learner about their experience and it was clear that they felt that the tool is able to achieve the declared objectives. In an effort to provide an evaluation of the impact of this learning approach, both from a self-reflection and awareness point of view, and of its instructional effectiveness, we ran some autonomous experiments, each devoted to stress one aspect of the expected impact of the tool developed. First, we carried out a set of two test cases devoted to explore the impact of an adaptive externalisation of the learner profiles. In this context, we collected data using structured questionnaires. The evaluation of the quantitative answers seems promising, even if some minor problematic aspects still exists. A parallel experiment explored the feasibility and foreseen impact generated by the mash-up of information from heterogeneous and distributed sources: in this case, the evaluation consisted of a simple, informal chat to the people that decided to participate to the study and who provided their personal navigational information. Finally, the last experiment we designed was devoted to explore the motivational effect of providing a social visualisation for autonomous learners in the context of extra-curricular courses. From this experience a possible increase in the commitment and in the outcome result could be seen, as measured by self-evaluation quizzes.

To sum things up, the contribution of this work is twofold: in the orchestration domain on the one hand, and in the support for the sustainability of TEL solutions on the other:

- On the former, three concurrent factors could positively influence the learner experience: the adaptability, in order to reduce the cognitive overload required for the information interpretation, the support of meta-cognitive skills through self-reflection processes operated by the learners and the engagement induced by the social aspects supported by the tool.

- On the latter, an enhancement of the awareness around the learning experience through the visualisation for presenting information can play a major role in supporting the tutor duties and enabling teachers to improve the design of the learning experience and activities, based on real usage data and learners interactions.

We are aware of the possibility to achieve negative effects through the application of our infrastructure to TEL experiences, which range from unexpected behaviors in groups with a low level of participation or riding phenomenons[1], to the impact of this data on learners with a less active attitude to interact and collaborate and to the informative overload for newly comers. We will present these aspects and possible directions for facing them in the next section.

[1]It is defined *riding* the phenomenon in which one or more persons use and take advantage of the work of someone else without providing any contribution and claiming that as collective/group shared work.

6.3 Future Possible Enhancements

One of the main issues concerned with the creation of a truly usable tool regards the provision of an easy and intuitive interface to set parameters and write configurations. The absence of an editor for the XML configuration layers for the specification of the semantics and operations to be applied to the data is in fact an open argument. The idea of integrating the editing interface with the GUI of the GRAPPLE Authoring Toolkit (GAT) is a possibility. Due to time limit and formal end point of the project, it was decided not to explore this direction. The decision was reinforced by the fact that GVIS is designed to be used with any kind of data source and in any kind of educational infrastructure/environment, making the integration with a specific environment a less than ideal solution. A possible improvement (which can be substantial for the real adoption and spread of the solution across different and heterogeneous environments, like the one characterising a PLE) is the integration of a graphical editor. Prerequisite for this step is the creation of a graphical language to encode the single operations of extraction, aggregation and encoding of pieces of information into widgets, based on the *DERI Pipes* project at http://pipes.deri.org (Morbidoni, Polleres, Tummarello, and Le-Phuoc 2007), (Phuoc, Polleres, Tummarello, and Morbidoni 2008), and (Daniel and Matera 2009), where a web-based interface was developed to connect and aggregate semantic sources from the Internet through a smart graphical interface based on gas pipes metaphors.

Another possible improvement on the interface issue would consist in providing more filtering and data reordering procedures through an easy visual interface to facilitate the exploratory navigation of the information.

Another open point is the creation of a repository to store common and useful configurations, from which new users would especially benefit. This will in fact reduce the time required to familiarise with the instrument and allow the newbie to achieve some initial results in a shorter time-frame. Considering that this task would normally be considered as complementary to the main task of the Instructional Designers or teachers, it is crucial to make sure the learning process takes as little as possible of their time and focus. The idea is that once they quickly familiarise with, these users will be able to develop their usage skills on a step-to-step basis and eventually be able to effortlessly create their own model and configurations. This kind of approach has also emerged from the system users feedback as being the preferred one.

Providing a set of adaptation templates could simplify the usage of these capabilities by the Instruction Designers and also give them the seeds for thinking about the possible impact the system can play in the context of their own educational model and experiences.

Another improvement would see a more extensive and structured testing of the tool, both to understand its full potential and threats and to analyse more in depth the impact that a visualisation (in all its form: adaptive, social and others) can have on different type of education models, from blended courses to completely online ones or from single course to fully online degree.

While the above could open up interesting and new perspectives, it is beyond the scope of the present work and will therefore not be explored further.

Appendix A

XML Schema

In this appendix, the Schema for the XML configuration of the three levels are included for reference. For the explanation of the meaning of the fields included, please refer to the chapter about the Tool implementation.

extraction_schema.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://vmelab10.usilu.net/GVIS/extractor"
xmlns="http://vmelab10.usilu.net/GVIS/extractor"
elementFormDefault="qualified">

<xs:element name="extraction">
<xs:complexType>
<xs:sequence>
<xs:element ref="source" maxOccurs="unbounded" minOccurs="1" />
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="source" type="elementBlock" />

<xs:complexType name="elementBlock">
<xs:sequence>
<xs:element ref="accessinfo" maxOccurs="1" minOccurs="1" />
<xs:element ref="sensibility" maxOccurs="1" minOccurs="0" />
<xs:element ref="query" minOccurs="1" maxOccurs="unbounded" />
<xs:element ref="resulttype" />
</xs:sequence>
```

```
<xs:attribute name="name" type="xs:string" />
</xs:complexType>

<!-- information about the access point -->
<xs:element name="accessinfo" type="accessinfoT" />
<xs:complexType name="accessinfoT">
<xs:sequence>
<xs:element ref="accesstype" maxOccurs="1" minOccurs="1" />
<xs:element ref="accesspoint" maxOccurs="1" minOccurs="1" />
<xs:element ref="accessmode" maxOccurs="1" minOccurs="0" />
<xs:element ref="accesssource" maxOccurs="1" minOccurs="0" />
<xs:element ref="username" maxOccurs="1" minOccurs="0" />
<xs:element ref="password" maxOccurs="1" minOccurs="0" />
<xs:element ref="lifetime" maxOccurs="1" minOccurs="1" />
</xs:sequence>
</xs:complexType>

<!-- the data-storage type i.e. DBMS or WebService -->
<xs:element name="accesstype" type="xs:string" />
<!-- the server instance where to get the data -->
<xs:element name="accesspoint" type="xs:string" />
<xs:element name="accessmode" type="xs:string" />
<!-- the database where to get the data -->
<xs:element name="accesssource" type="xs:string" />
<!-- username to connect to the DB -->
    <xs:element name="username" type="xs:string" />
<!-- password to connect to the DB -->
    <xs:element name="password" type="xs:string" />
<!-- the source lifetime before refresh data -->
<xs:element name="lifetime" type="xs:integer" />


<xs:element name="sensibility" type="sensibilityList" />
<xs:complexType name="sensibilityList">
<xs:sequence>
<xs:element ref="filter" minOccurs="1" maxOccurs="unbounded" />
</xs:sequence>
</xs:complexType>

<xs:element name="filter" type="filterT" />
<xs:complexType name="filterT">
```

```
<xs:simpleContent>
<xs:extension base="xs:string">
<xs:attribute name="name" use="required">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:enumeration value="concept" />
<xs:enumeration value="activity" />
<xs:enumeration value="user" />
<xs:enumeration value="course" />
</xs:restriction>
</xs:simpleType>
</xs:attribute>
</xs:extension>
</xs:simpleContent>
</xs:complexType>

<xs:element name="query" type="queryT" />
<xs:complexType name="queryT">
<xs:sequence>
<xs:element ref="sql" maxOccurs="1" minOccurs="0" />
<xs:element ref="operation" maxOccurs="1" minOccurs="0" />
<xs:element ref="parameters" maxOccurs="1" minOccurs="0" />
<xs:element ref="normalize" minOccurs="0" maxOccurs="1" />
<xs:element ref="resulttype" />
</xs:sequence>
</xs:complexType>

<!-- the sql string -->
<xs:element name="sql" type="xs:string" />

<!-- the operation structure : WS querying -->
<xs:element name="operation" type='operationT' />
<xs:complexType name="operationT">
<xs:sequence>
<xs:element ref="function" minOccurs="1" maxOccurs="1" />
<xs:element name="header" type="headerT" minOccurs="0" maxOccurs="1" />
<xs:element ref="request" minOccurs="1" maxOccurs="1" />
</xs:sequence>
</xs:complexType>

<!-- function to call in WS invocation -->
```

```
<xs:element name="function" type="xs:string" />

<xs:complexType name="headerT">
<xs:sequence>
<xs:element name="h" type="spaceT" minOccurs="1"
maxOccurs="unbounded" />
<xs:element name="s" type="xs:string"></xs:element>
</xs:sequence>
</xs:complexType>

<xs:complexType name="spaceT">
<xs:sequence>
<xs:element name="pref" maxOccurs="1" minOccurs="1" type="xs:string"/>
<xs:element name="space" maxOccurs="1" minOccurs="1" type="xs:string"/>
</xs:sequence>
</xs:complexType>


<!-- parameters for calling WS -->
<xs:element name="request" type="xs:string" />

<!-- the dynamic part of WHERE clause -->
<xs:element name="parameters">
<xs:complexType>
<xs:sequence>
<xs:choice minOccurs="1" maxOccurs="unbounded">
<xs:element ref="param" minOccurs="1" maxOccurs="unbounded" />
</xs:choice>
</xs:sequence>
</xs:complexType>
</xs:element>

<!-- name of dynamic value from an other level for the WHERE clause -->
<xs:element name="param" type="paramT" />
<xs:complexType name="paramT">
<xs:simpleContent>
<xs:extension base="xs:string">
<xs:attribute use="optional" name="prevquery" type="xs:boolean"
default="true" />
</xs:extension>
</xs:simpleContent>
```

```
</xs:complexType>

<!-- normalization of the result is needed by which value -->
<xs:element name="normalize" type="xs:string" />

<!-- the returned pseudo type of that extraction -->
<xs:element name="resulttype">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:enumeration value="numeric" />
<xs:enumeration value="string" />
<xs:enumeration value="list" />
<xs:enumeration value="record" />
<xs:enumeration value="listofrecords" />
</xs:restriction>
</xs:simpleType>
</xs:element>

</xs:schema>
```

aggregation_schema.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://vmelab10.usilu.net/GVIS/aggregator"
xmlns="http://vmelab10.usilu.net/GVIS/aggregator"
elementFormDefault="qualified">

<xs:element name="aggregation">
<xs:complexType>
<xs:sequence>
<xs:element ref="source" maxOccurs="unbounded" minOccurs="1" />
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="source" type="elementBlock" />
<xs:complexType name="elementBlock">
<xs:sequence>
<xs:element ref="extraction" maxOccurs="1" minOccurs="1" />
<xs:element ref="computation" maxOccurs="1" minOccurs="0" />
<xs:element ref="resulttype" maxOccurs="1" minOccurs="1" />
```

```
</xs:sequence>
<xs:attribute name="name" type="xs:string" />
</xs:complexType>

<!-- extraction resources block -->
<xs:element name="extraction" type="extractionT" />
<xs:complexType name="extractionT">
<xs:sequence>
<xs:element ref="toextract" maxOccurs="unbounded"
minOccurs="1" />
</xs:sequence>
</xs:complexType>

<!-- aggregations computation block -->
<xs:element name="computation" type="computationT" />
<xs:complexType name="computationT">
<xs:sequence>
<xs:element ref="tocompute" maxOccurs="unbounded"
minOccurs="1" />
</xs:sequence>
</xs:complexType>

<!-- name of extraction block -->
<xs:element name="toextract" type="extractT" />
<xs:complexType name="extractT">
<xs:simpleContent>
<xs:extension base="xs:string">
<!--
this attribute defines if the extraction has to be performed
immediatly or not
-->
<xs:attribute name="defer" default="false" use="optional">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:enumeration value="true" />
<xs:enumeration value="false" />
</xs:restriction>
</xs:simpleType>
</xs:attribute>
</xs:extension>
</xs:simpleContent>
```

```
</xs:complexType>

<!-- what to do after the extraction -->
<xs:element name="tocompute" type="computeT" />
<xs:complexType name="computeT">
<xs:sequence>
<xs:element ref="operation" maxOccurs="1" minOccurs="1" />
<xs:element ref="parameters" maxOccurs="1" minOccurs="0" />
<xs:element ref="resulttype" maxOccurs="1" minOccurs="1" />
</xs:sequence>
</xs:complexType>

<!-- an existent operation in the command-pattern -->
<xs:element name="operation" type="xs:string" />

<xs:element name="parameters">
<xs:complexType>
<xs:sequence>
<xs:element ref="param" maxOccurs="unbounded" minOccurs="1" />
</xs:sequence>
</xs:complexType>
</xs:element>

<!-- the number of extracted element from zero -->
<xs:element name="param" type="paramT" />
<xs:complexType name="paramT">
<xs:simpleContent>
<xs:extension base="xs:string">
<xs:attribute use="optional" name="computed" type="xs:boolean"
default="true" />
</xs:extension>
</xs:simpleContent>
</xs:complexType>

<!-- the returned pseudo type of that aggregation -->
<xs:element name="resulttype">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:enumeration value="numeric" />
<xs:enumeration value="string" />
<xs:enumeration value="list" />
```

```
<xs:enumeration value="record" />
<xs:enumeration value="listofrecords" />
</xs:restriction>
</xs:simpleType>
</xs:element>

</xs:schema>
```

building_schema.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://vmelab10.usilu.net/GVIS/builder" xmlns="http://vmelab10.usilu.net/GVIS/build
elementFormDefault="qualified">

<xs:element name="building">
<xs:complexType>
<xs:sequence>
<xs:element ref="widget" maxOccurs="unbounded" minOccurs="1" />
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="widget" type="elementBlock" />
<xs:complexType name="elementBlock">
<xs:sequence>
<xs:element ref="properties" maxOccurs="1" minOccurs="1" />
<xs:element ref="chart" maxOccurs="1" minOccurs="1" />
</xs:sequence>
<xs:attribute name="name" type="xs:string" />
</xs:complexType>

<!-- widget static properties -->
<xs:element name="properties" type="propertiesT" />
<xs:complexType name="propertiesT">
<xs:sequence>
<xs:element ref="title" maxOccurs="1" minOccurs="1" />
<xs:element ref="backcolor" maxOccurs="1" minOccurs="1" />
<xs:element ref="elementcolor" maxOccurs="1" minOccurs="1" />
<xs:element ref="xaxis" maxOccurs="1" minOccurs="0" />
<xs:element ref="yaxis" maxOccurs="1" minOccurs="0" />
</xs:sequence>
```

```xml
</xs:complexType>

<!-- displayed title -->
<xs:element name="title" type="xs:string" />

<!-- the color of the widget's background -->
<xs:element name="backcolor" type="webcolor" />

<!-- the color of the chart elements -->
<xs:element name="elementcolor" type="webcolor" />

<xs:simpleType name="webcolor">
<xs:restriction base="xs:string">
<xs:length value="7" />
<xs:pattern value="#([a-fA-F0-9]){6}" />
</xs:restriction>
</xs:simpleType>

<!-- axes labels -->
<xs:element name="xaxis" type="xs:string" />
<xs:element name="yaxis" type="xs:string" />

<!-- elements block to insert into the chart -->
<xs:element name="chart" type="chartT" />
<xs:complexType name="chartT">
<xs:sequence>
<xs:element ref="chartsource" minOccurs="1" maxOccurs="unbounded" />
</xs:sequence>
<!-- the type of the chart such as: bar, pie, scatter and so on -->
<xs:attribute name="type">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:enumeration value="bar" />
<xs:enumeration value="hbar" />
<xs:enumeration value="pie" />
<xs:enumeration value="scatter" />
<!-- This tell to the builder code this widget is not graphic -->
<xs:enumeration value="text" />
</xs:restriction>
</xs:simpleType>
</xs:attribute>
```

```
</xs:complexType>

<!-- block to get data and relative association -->
<xs:element name="chartsource" type="chartsourceT" />
<xs:complexType name="chartsourceT">
<xs:sequence>
<xs:element ref="data" maxOccurs="1" minOccurs="1" />
<xs:element ref="mapping" maxOccurs="1" minOccurs="1" />
<xs:element ref="label" maxOccurs="1" minOccurs="0" />
<xs:element ref="elementcolor" maxOccurs="1" minOccurs="0" />
</xs:sequence>
</xs:complexType>

<!-- data from the aggregation layer -->
<xs:element name="data" type="dataT" />
<xs:complexType name="dataT">
<xs:simpleContent>
<xs:extension base="xs:string">
<xs:attribute name="from">
<xs:simpleType>
<xs:restriction base="xs:string">
<!-- This means the builder gets data from aggregation -->
<xs:enumeration value="system" />
<!-- This means the builder gets data from session environement -->
<xs:enumeration value="session" />
<!-- This means the builder will use the string in the setting -->
<xs:enumeration value="string" />
</xs:restriction>
</xs:simpleType>
</xs:attribute>
</xs:extension>
</xs:simpleContent>
</xs:complexType>

<!-- mapping of data to graphical element -->
<xs:element name="mapping">
<xs:simpleType>
<xs:restriction base="xs:string">
<!-- This means map to chart elements -->
<xs:enumeration value="values" />
<!-- needed by Pie chart -->
```

```
<xs:enumeration value="total" />
<!-- This means map as a second chart values -->
<xs:enumeration value="comparison" />
<xs:enumeration value="string" />
<xs:enumeration value="number" />
</xs:restriction>
</xs:simpleType>
</xs:element>

<!-- The label needed by eventually single value results -->
<xs:element name="label" type="xs:string" />

</xs:schema>
```

Appendix B

JSON messages

In this appendix, the structure of the JSON messages returned from the GVIS architecture to the instance of the dashboard for the final composition of the widgets is presented. For the meaning of the fields included, please refer to the chapter about the Tool implementation.

```
+ROOT
|
| +ARRAY: "elements"
| |
| | -STRING: "type"
| |
| | -BOOLEAN: "gradient-fill"
| |
| | +ARRAY: "animate"
| | |
| | | -STRING: "type"
| | |
| | | -INTEGER: "distance"
| |
| | +ARRAY: "values"
| | |
| | | - INTEGER: "value"
| | |
| | | - STRING: "label"
| | |
| | | - STRING: "colour"
| |
| | - STRING: "tip"
|
```

```
| + OBJ: "title"
| |
| | -STRING: "text"
|
| - STRING: "bg_color"
|
```

And an example of the JSON message used can be the following one:


```
{
  "elements":[
    {
      "type":"pie",
      "gradient-fill":true,
      "animate":[
        {
          "type":"bounce",
          "distance":10
        }
     ],
     "values":[
        {
          "value":142,
          "label":"XXX",
          "colour":"00AA00"
        },
...
...
       {
          "value":19,
          "label":"YYY",
          "colour":"AA00AA"
        },
       {
          "value":10,
          "label":"ZZZ",
          "colour":"00AAAA"
        }
     ],
     "tip":"#val# - #percent#"
    }
```

```
 ],
 "title":{
    "text":"Delicious Keywords (clustered) about EgidioM firefox History"
  },
  "bg_color": "#FFFFFF"
}
```

References

[1] Abel, F., Henze, N., Herder, E., Krause, D. (2010). Interweaving Public User Profiles on the Web. Proc. UMAP 2010.

[2] Abel, F., Herder, E., Houben, G.J., Henze, N. and Krause, D. (2010). Cross-system User Modeling and Personalization on the Social Web. UMUAI - User Modeling and User-Adapted Interaction.

[3] Agrawal, R., Imieliński, T., and Swami, A. (1993). Mining association rules between sets of items in large databases. In ACM SIGMOD Record (Vol. 22, No. 2, pp. 207-216). ACM.

[4] Ahmad, A.R., Basir, O.A., Hassanein, K. (2004). Adaptive User Interfaces for Intelligent E-Learning: Issues and Trends. The Fourth International Conference on Electronic Business - Shaping Business Strategy in a Networked World - ICEB 2004: pp. 925-934

[5] Ahmad, N., Bull, S. (2009). Learner Trust in Learner Model Externalisations, in V. Dimitrova, R. Mizoguchi, B. du Boulay & A. Graesser (eds), Artificial Intelligence in Education 2009, IOS Press, Amsterdam, 617-619.

[6] Arnold, K.E., and Pistilli, M.D. (2012). Course signals at Purdue: using learning analytics to increase student success. In Proceedings of the 2nd International Conference on Learning Analytics and Knowledge (LAK '12), Simon Buckingham Shum, Dragan Gasevic, and Rebecca Ferguson (Eds.). ACM, New York, NY, USA, pgg. 267 - 270. DOI=10.1145/2330601.2330666 http://doi.acm.org/10.1145/2330601.2330666

[7] Arroyo, I., Ferguson, K., Johns, J., Dragon, T., Mehranian, H., Fisher, D., Barto, A.,Mahadevan, S., Woolf, B. (2007). Repairing Disengagement with Non Invasive Interventions. In: International Conference on Artificial Intelligence in Education, Marina del Rey, CA, pp. 195-202

[8] Baker, R. (2010). Data Mining for Education. In McGaw, B., Peterson, P., Baker, E. (Eds.) International Encyclopedia of Education (3rd edition), vol. 7, pp. 112-118. Oxford, UK: Elsevier.

[9] Bandura, A. (1997). Self−efficacy: The exercise of control. New York: Freeman.

[10] Bienkowski, M., Feng, M., Means, B. (2012). Enhancing Teaching and Learning Through Educational Data Mining and Learning Analytics. U.S. Department of Education - Office of Educational Technology.

[11] Suh, B., Chi, E.H., Kittur, A., Pendleton, B.A. (2008). Lifting the veil: improving accountability and social transparency in Wikipedia with wikidashboard. In Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems (CHI '08). ACM, New York, NY, USA, 1037-1040. DOI=10.1145/1357054.1357214 http://doi.acm.org/10.1145/1357054.1357214

[12] Brusilovsky, P. (2001). WebEx: Learning from examples in a programming course. In: Fowler, W. and Hasebrook, J. (eds.) Proc. of WebNet'2001, World Conference of the WWW and Internet, Orlando, FL, AACE, 124-129.

[13] Brusilovsky, P. (2004). KnowledgeTree: A distributed architecture for adaptive e-learning. In: Proceedings of The Thirteenth International World Wide Web Conference, WWW 2004 (Alternate track papers and posters), New York, NY, 17-22, ACM Press, pp. 104-113.

[14] Brusilovsky, P., and Peylo, C., (2003). Adaptive and intelligent Web-based educational systems. In International Journal of Artificial Intelligence in Education volume 13 Issue 2-4, April 2003, Pages 159-172. IOS Press Amsterdam, The Netherlands, The Netherlands

[15] Brusilovsky, P., Sosnovsky, S., and Shcherbinina, O. (2004). QuizGuide: Increasing the educational value of individualized selfassessment quizzes with adaptive navigation support. In: J. Nall and R. Robson (Eds.). Proceedings of World Conference on E-Learning, E-Learn 2004, Washington, DC, USA, November 1-5, 2004, AACE, pp. 1806-1813.

[16] Bull, S (1997). See Yourself Write: A Simple Student Model to Make Students Think. In Jameson, A., Paris, C., and Tasso, C., editors, User Modelling (Proceedings from 6th International Conference, UM 1997), pp. 315-326.

[17] Bull, S. (2004). Supporting learning with open learner models. In: Proceedings of 4th Hellenic Conference on Information and Communication Technologies in Education, Athens, Greece,September 29 - October 3, 2004, pp. 47-61

[18] Bull, S. and Britland, M. (2007). Group Interaction Prompted by a Simple Assessed Open Learner Model that can be Optionally Released to Peers, in P. Brusilovsky, M. Grigoriadou and K. Papanikolaou (eds), Proceedings of Workshop on Personalisation in E-Learning Environments at Individual and Group Level, User Modeling 2007.

[19] Bull, S., Dimitrova, V, McCalla, G. , eds (2007). International Journal of Artificial Intelligence in Education 17(2) 2007, Special Issue (Part 1) "Open Learner Models: Research Questions" (2007a)

[20] Bull, S., Dimitrova, V, McCalla, G. , eds (2007). International Journal of Artificial Intelligence in Education 17(2) 2007, Special Issue (Part 2) "Open Learner Models: Future Research Directions" (2007b)

[21] Bull, S., Gardner, P., Ahmad, N., Ting, J., and Clarke, B. (2009). Use and Trust of Simple Independent Open Learner Models to Support Learning within and across Courses. LNCS, vol. 5535. Springer-Verlag, Berlin, Heidelberg, pp. 42-53.

[22] Bull, S., Greer, J. and McCalla, G. (2003). The Caring Personal Agent, International Journal of Artificial Intelligence in Education 13(1), 21-34.

[23] Bull, S., and Kay, J. (2007). Student Models that Invite the Learner In: The SMILI :() Open Learner Modelling Framework. International Journal of Artificial Intelligence in Education, 17(2). pgg. 89-120

[24] Bull, S., Kay, J. (2008). Metacognition and Open Learner Models. In Metacognition Workshop, Intelligent Tutoring Systems 2008, ppg. 7 - 20.

[25] Bull, S., and Nghiem, T. (2002). Helping Learners to Understand Themselves with a Learner Model Open to Students, Peers and Instructors. Paper presented at the International Conference on Intelligent Tutoring Systems 2002 - Workshop on Individual and Group Modelling Methods, pgg. 5 - 13.

[26] Card, S. K., Mackinlay, J. D., and Schneiderman, B. (Eds.). (1999). Readings in information visualization: using vision to think. Morgan Kaufmann.

[27] Chen, C. (2009). Influence of Perceived Information Overload on Learning in Computer-Mediated Communication. In Proceedings of the 8th international Conference on Advances in Web Based Learning, pgg. 112 - 115.

[28] Chen, Z.H., Chou, C.Y., Deng, Y.C., and Chan, T.W. (2007). Active Open Learner Models as Animal Companions: Motivating Children to Learn through Interaction with My-Pet and Our-Pet, International Journal of Artificial Intelligence in Education, 17(3): 217-226

[29] Chi, E. H., Suh, B., Kittur, A. (2008). Providing social transparency through visualizations in Wikipedia. Social Data Analysis Workshop at CHI2008

[30] Conlan O., O'Keeffe I., Brady A., and Wade V. (2007). Principles for Designing Activity-based Personalized eLearningIEEE International Conference on Advanced Learning Technologies, pp. 642-644, Seventh IEEE International Conference on Advanced Learning Technologies (ICALT 2007).

[31] Costabile, M.F., De Marsico, M., Lanzilotti, R., Plantamura, V.L., and Roselli, T. (2005). On the Usability Evaluation of E-Learning Applications. In Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05) - Track 1 - Volume 01. IEEE Computer Society, Washington, DC, USA, 6.2-. DOI=10.1109/HICSS.2005.468 Available online at http://dx.doi.org/10.1109/HICSS.2005.468

[32] Courtin, C., Talbot, S. (2005). An Architecture to Record Traces in Instrumented Collaborative Learning Environments. CELDA 2005, pp. 301-308. Available online at http://www.iadisportal.org/digital-library/mdownload/an-architecture-to-record-traces-in-instrumented-collaborative-learning-environments

[33] Courtin, C., Talbot, S. (2006). Trace Analysis in Instrumented Collaborative Learning Environments. 6th International Conference on Advanced Learning Technologies, pp.1036-1038. doi: 10.1109/ICALT.2006.1652623 Available online at http://ieeexplore.ieee.org/stamp/stamp.jsp? tp=&arnumber=1652623&isnumber=34637

[34] Daniel, F. and Matera, M. (2009). Turning Web Applications into Mashupbrusli Components: Issues, Models, and Solutions. In: Gaedke, Martin and Grossniklaus, Michael and Daz, Oscar (eds.) Web Engineering - Lecture Notes in Computer Science, Springer Berlin Heidelberg, vol. 5648, pp. 45-60 ISBN: 978-3-642-02817-5 DOI: 10.1007/978-3-642-02818-2_4

[35] Dattolo, A., Eynard, D., and Mazzola, L. (2011). An Integrated Approach to Discover Tag Semantics. 26th Annual ACM Symposium on Applied Computing, vol. 1, pp. 814-820. Taichung, Taiwan, March 2001.

[36] Dawson, S. (2010). Seeing the learning community: An exploration of the development of a resource for monitoring online student networking. British Journal of Educational Technology,41(5), pgg. 736-752

[37] Dawson, S., Bakharia, A., and Heathcote, E. (2010). SNAPP: Realizing the affordances of real-time SNA within networked learning environments. In Proceedings of the 7th International Conference on Networked Learning pp. 125-133.

[38] De Bra, P., Pechenizkiy, M., van der Sluijs, K. and Smits, D. (2008). GRAPPLE: Integrating Adaptive Learning into Learning Management Systems.In Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2008 (pp. 5183-5188). Chesapeake, VA: AACE.

[39] Dimitrova, V., (2003). STyLE-OLM: Interactive Open Learner Modelling. Int. J. Artif. Intell. Ed. 13, 1, pgg. 35 - 78.

[40] Dimitrova, V., Self, J., Brna, P. (2001). Applying Interactive Open Learner Models to Learning Technical Terminology. Proceedings of User Modeling'2001, Springer, pp. 148-157.

[41] Dolog, P., Henze, N., Nejdl, W., and Sintek, M. (2004). Personalization in distributed e-learning environments.In Proceedings of the 13th international World Wide Web Conference on Alternate Track Papers & Posters (New York, NY, USA, May 19 - 21, 2004). WWW Alt. '04. ACM, New York, NY, pp. 170-179.

[42] Dror, B., Nadine, M., Mike, B., (2008). Visualization and Analysis of Student Interaction in an Adaptive Exploratory Learning Environment. In Proceedings of 1st Int. Workshop in Intelligent Support for Exploratory Environments, EC-TEL'08, pgg. 1 - 10.

[43] AA.VV. "EducationalDataMining.org". 2010. http://www.educationaldatamining.org/. Retrieved 2011-01-16.

[44] Erickson, T., Kellogg, W.A. (2000). Social translucence: an approach to designing systems that support social processes. ACM Transactions on Computer-Human Interaction. vol. 7 (1), pp. 59-83. DOI=10.1145/344949.345004 http://doi.acm.org/10.1145/344949.345004

[45] Erickson, T. and Kellogg, W.A. (2001). Social translucence: Designing systems that support social processes. In J.M. Carroll (Ed.), Human-computer interaction in the new millennium. NY: Addison-Wesley.

[46] Erickson, T., Halverson, C., Kellogg, W.A., Laff, M., and Wolf, T. (2002). Social translucence: designing social infrastructures that make collective activity visible. Commununication of the ACM 45, 4 (April 2002), pp. 40-44. DOI=10.1145/505248.505270 http://doi.acm.org/10.1145/505248.505270

[47] Esposito, F., Licchelli, O. and Semeraro, G. (2004). Discovering Student Models in e-learning Systems. In Journal of Universal Computer Science, vol. 10, no. 1 (2004), 47-57 . Availabe online: http://dx.doi.org/10.3217/jucs-010-01-0047

[48] Eynard, D., Mazzola, L., Dattolo, A. (2012). Exploiting tag similarities to discover synonyms and homonyms in folksonomies. Software: Practice and Experience. Wiley & Sons, Ltd. doi: 10.1002/spe.2150

[49] Falakmasir, M., Hsiao, I., Mazzola, L., Grant, N., Brusilovsky P. (2012). The Impact of Social Performance Visualization on Students. Proceedings of the 12th IEEE International Conference on Advanced Learning Technologies and Technology-Enhanced Learning, Rome, Italy, July 4-6, 2012 pgg. 565-569.

[50] Ferguson, R. (2012). The State Of Learning Analytics in 2012: A Review and Future Challenges. Technical Report KMI-12-01, Knowledge Media Institute, The Open University, UK. http://kmi.open.ac.uk/publications/ techreport/kmi-12-01

[51] Ferreira de Oliveira, M.C., Levkowitz, H. (2003). From visual data exploration to visual data mining: a survey. IEEE Transactions on Visualization and Computer Graphics, vol.9, no.3, pp. 378- 394, July-Sept. 2003 DOI: 10.1109/TVCG.2003.1207445 Available online at http://ieeexplore.ieee.org/stamp/stamp.jsp? tp=&arnumber=1207445&isnumber=27179

[52] Fischer, G. (2001). User Modeling in Human-Computer Interaction. In User Modeling and User-Adapted Interaction 11, 1-2 (Mar. 2001), pgg 65-86.

[53] Gama, C. (2004). Metacognition in Interactive Learning Environments: The Reflection Assistant Model. Intelligent Tutoring Systems, Springer-Verlag, Berlin Heidelberg, 668-677.

[54] Glahn, C. (2009). Contextual support of social engagement and reflection on the Web. Doctoral thesis. September, 18, 2009, Heerlen, The Netherlands: Open University of the Netherlands, CELSTEC. URI: http://hdl.handle.net/1820/2062 ISBN: 978-90-79447-26-8

[55] Glahn, C.; Specht, M.; Koper, R. (2008). Smart indicators to support the learning interaction cycle. International Journal of Continuing Engineering Education and Lifelong Learning, Vol. 18, No. 1.

[56] Gediga, G., Hamborg, K.-C., and Düntsch, I., (1999). The IsoMetrics Usability Inventory: An operationalisation of ISO 9241-10. Behaviour and Information Technology, 18, 151-164.

[57] Graf, S., and Kinshuk (2008). Learner Modelling Through Analyzing Cognitive Skills and Learning Styles. In Adelsberger, H., Kinshuk, P., Pawlowski, J., and Sampson, D. (Eds.) Handbook on Information Technologies for Education and Training. 2nd.Springer Publishing Company, Incorporated. Pp 179-194.

[58] Graf, S., Ives, C., Rahman, N., Ferri, A. (2011). AAT - A Tool for Accessing and Analysing Students' Behavior Data in Learning Systems. In Proceedings of the 1st International Conference on Learning Analytics and Knowledge (LAK '11). ACM, New York, NY, USA, pgg. 174 - 179. http://doi.acm.org/10.1145/2090116.2090145

[59] GRAPPLE project available online at: http://grapple-project.org

[60] Gwizdka, J., Bakelaar, P. (2009). Tag trails: navigation with context and history. In Proceedings of the 27th international Conference Extended Abstracts on Human Factors in Computing Systems (Boston, MA, USA, April 04 - 09, 2009). CHI EA '09. ACM, New York, NY, 4579-4584.

[61] Jeff, T. (2005). Software quality engineering: testing, quality assurance, and quantifiable Improvements. John Wiley

[62] Kay, J. (2006). Scrutable Adaptation: Because We Can and Must. 2006, pp. 11-19.

[63] Kay, J. (2008). Life-Long Learning, Learner Models and Augmented Cognition. Intelligent Tutoring Systems 2008, pp. 3-5.

[64] Kay, J., Kummerfeld, B. (2010). Tackling HCI Challenges of Creating Personalised, Pervasive Learning Ecosystems. EC-TEL 2010, pp. 1-16

[65] Kay, J., Reimann, P., Yacef, K. (2007). Mirroring of Group Activity to Support Learning as Participation. AIED 2007, pp. 584-586.

[66] Kehrwald, B. (2010). Social Presence and Online Communication: A response to Mersham. Journal of Open, Flexible, and Distance Learning, 2010, 14(1), pp. 29 - 46. Available online at http://journals.akoaotearoa.ac.nz/index.php/ JOFDL/article/view/28/25

[67] Kroski, E. (2005). The hive mind: Folksonomies and user-based tagging, December 2005. http://infotangle.blogsome.com/2005/12/07/the-hive-mind-folksonomies-and-user-based-tagging/.

[68] Haag, S., Cummings, M., and Phillips, A. (2007). Management Information Systems for the Information Age.6th. McGraw-Hill Irwin.

[69] Halpin, H., Robu, V., Shepard, H. (2006). The dynamics and semantics of collaborative tagging. In Proceedings of the 1st Semantic Authoring and Annotation Workshop (SAAW06), 2006.

[70] Harrigan, M., Kravcik, M., Steiner, C., and Wade, V. (2009). Requirements Specification by Academic Users on Integrated Adaptive. Grapple D9.1 Deliverable, Version, 1.

[71] Heath, T., Motta, E., and Dzbor, M. (2005). Uses of Contextual Information to Support Online Tasks.In Proceedings 1st AKT Doctoral Symposium, pages pp. 107-113, KMi, The Open University, Milton Keynes.

[72] Heckmann, D., Schwartz, T., Brandherm, B., Kröner , A., (2005). Decentralized User Modeling with UserML and GUMO. Proceedings of the Workshop on Decentralized, Agent Based and Social Approaches to User Modelling (DASUM 2005), Edinburgh, Scotland, pp. 61-65

[73] Heath, T., Motta, E., Dzbor, M. (2005). Uses of Contextual Information to Support Online Tasks. In Proceedings 1st AKT Doctoral Symposium, pages pp. 107-113, KMi, The Open University, Milton Keynes.

[74] Hsiao, I.-H., Bakalov, F., Brusilovsky, P., Konig-Ries, B. (2011). Open Social Student Modeling: Visualizing Student Models with Parallel Introspective Views. In: Konstan, J.A., Conejo, R., Marzo, J.L., Oliver, N. (eds.) UMAP 2011. LNCS, vol. 6787, pp. 171 - 182. Springer, Heidelberg

[75] Hsiao, I.-H., Brusilovsky, P. (2012). Motivational Social Visualizations for Personalized E-Learning. 21st Century Learning for 21st Century Skills - Lecture Notes in Computer Science, 2012, Volume 7563/2012, pp. 153-165, DOI: 10.1007/978-3-642-33263-0_13

[76] Hoppe, U., Ogata, H., and Soller, A. (2007). The Role of Technology in Cscl: Studies in Technology Enhanced Collaborative Learning. Springer

[77] Laurillard, D., Oliver, M., Wasson, B., Ulrich, H. (2009). Implementing Technology-Enhanced Learning. In: Technology-Enhanced Learning: Principles and Products. Springer-Link, Netherlands, pp. 289-306

[78] Laurillard, D., Oliver, M.,Wasson, B., Ulrich, H. (2009). Implementing Technology-Enhanced Learning. In: Technology-Enhanced Learning: Principles and Products. Springer-Link, Netherlands, pp. 289-306.

[79] Leonardi, E., Abel, F., Heckmann, D., Herder, E., Hidders, J., Houben, G.J. (2010). A Flexible Rule-Based Method for Interlinking, Integrating, and Enriching User Data. Proc. ICWE 2010

[80] Levy, Y. (2007). Comparing dropouts and persistence in e-learning courses. Computers & Education, Volume 48, Issue 2, February 2007, Pp. 185-204, ISSN 0360-1315, DOI: 10.1016/j.compedu.2004.12.004.

[81] Looi, C.K., McCalla, G., Bredeweg, B., Breuker, J. (2005). Artificial Intelligence in Education - Supporting Learning through Intelligent and Socially Informed Technology. Proceedings of the 12th International Conference on Artificial Intelligence in Education, AIED 2005, July 18-22, 2005, Amsterdam, The Netherlands - IOS Press

[82] Long, Y., Aleven, V. (2011). Students' Understanding of Their Student Model, In: International Journal of Artificial Intelligence in Education 67(38), 179-186.

[83] Mabbott, A., and Bull, S. (2004). Alternative Views on Knowledge: Presentation of Open Learner Models, in J.C. Lester, R.M. Vicari & F. Paraguacu (eds), Intelligent Tutoring Systems: 7th International Conference, Springer-Verlag, Berlin Heidelberg, pp. 689-698.

[84] Mabbott, A. and Bull, S. (2006). Student Preferences for Editing, Persuading and Negotiating the Open Learner Model. In: Intelligent Tutoring Systems, Springer-Verlag, Berlin Heidelberg, 481-490.

[85] Maglogiannis, I., Karpouzis, K., Bramer, M. (2009). Part II: Intelligent human-computer interaction. Special Issue on Emerging Artificial Intelligence Applications and Innovations: Papers from AIAI 2006. Appl. Intell. 30(1): pp. 1-2 (2009)

[86] Mahlke, S., Lindgaard, G. (2007). Emotional experiences and quality perceptions of interactive products.(pp. 164-173).

[87] May, M., George, S., Prevot, P. (2007). Tracking, Analyzing, and Visualizing Learners' Activities on Discussion Forums. Proc. International Conference on Web-based Education (WBE 2007), Chamonix, pp. 649-656.

[88] Mazza, R., Bettoni, M., Fare, M., Mazzola, L. (2012). MOCLog: Monitoring Online Courses with log data. In Retalis, S., and Dougiamas, M. Proceeding of the 1st Moodle Research Conference.

[89] Mazza, R., Botturi, L., Tardini, S. (Eds.) (2006). FOSLET 2006. Proceeding of Workshop on Free and Open Source Learning Environments and Tool. Como, Italy

[90] Mazza, R., Dimitrova, V. (2004). Visualising Student Tracking Data to Support Instructors in Web-Based Distance Education, 13th International World Wide Web Conference - Alternate Educational Track, pp. 154-161.

[91] Mazza, R. and Dimitrova, V. (2007). CourseVis: a graphical student monitoring tool for supporting instructors in web-based distancecourses. International Journal of Human-Computer Studies, 65, 2, pgg. 125-139.

[92] Mazza, R., Mazzola, L., Glahn, C., Verpoorten, D., Nussbaumer, A., Steiner, C., and Heckmann, D. (2009). Design of interactive visualization of models and students data. Grapple D4.5a Deliverable, Version, 2.

[93] Mazza, R., Mazzola, L., Glahn, C., Nussbaumer, A., and Verpoorten, D. (2010). Implementation of interactive visualization of models and students data. Grapple D4.5b Deliverable, Version, 1.

[94] Mazza, R., Mazzola, L., Glahn, C., Nussbaumer, A., and Verpoorten, D. (2011). Evaluation and refinement of interactive visualization of models and students data. Grapple D4.5c Deliverable, Version, 2.

[95] Mazza, R., Milani, C. (2004). GISMO: a Graphical Interactive Student Monitoring Tool for Course Management Systems. Proceedings of the International Conference: T.E.L.04 Technology Enhanced Learning 04. Milan, 18-19November 2004.

[96] Mazza, R., Milani, C. (2005). Exploring Usage Analysis in Learning Systems: Gaining Insights from Visualisations. Proceedings of the Conference: AIED05 International Conference on Artificial Intelligence in Education. Amsterdam, 18-22 July 2005.

[97] Mazzetti, A., Tenerini, P., Dicerto, M., Van der Sluijs, K., Smits, D., Rambout, D., Abel, F., Pekczynski, P., Mazza, R., Mazzola, L., Gaeremynck, S., Foss, J., Minne, F., and Vasilyeva, E. (2011). Final specification of the operational infrastructure. Grapple D7.1c Deliverable, Version, 1.

[98] Mazzola L., Eynard D., Mazza R. (2010). GVIS: a framework for graphical mashups of heterogeneous sources to support data interpretation. Human System Interactions, 2010. 3nd Conference on HSI '10. pgg. 578 - 584.

[99] Mazzola, L., Mazza, R. (2009). Supporting learners in Adaptive Learning Environments through the enhancement of the Student Model. In Proceeding of Human Computer Interaction International 2009, pgg. 166 - 175. San Diego, CA (2009a).

[100] Mazzola, L. and Mazza, R. (2009). Toward adaptive representations of student models in eLearning environments. In Proceeding of 14th International Conference on Artificial Intelligence in Education (AIED 2009), pgg. 761 - 762. Brighton, UK (2009b).

[101] Mazzola L., Mazza R. (2010). GVIS: A Facility for Adaptively Mashing Up and Representing Open Learner Models. 'Sustaining TEL: From Innovation to Learning and Practice' in Lecture Notes in Computer Science, Volume 6383/2010, pp. 554-559, Proceedings of EC-TEL 2010, DOI: 10.1007/978-3-642-16020-2_53 .

[102] Mazzola L., Mazza R. (2011). Visualizing Learner Models through data aggregation: a test case. . Red-conference, rethinking education in the knowledge society, Monte Veritá, Switzerland, March 7-10, pp.372- 380, ISBN 978-88-6101-010-9 .

[103] McCalla, G. (2004). The Ecological Approach to the Design of E-Learning Environments: Purpose-based Capture and Use of Information About Learners.In Journal of Interactive Media in Education, 2004 (7).

[104] Mitrovic, A., Martin, B. (2007). Evaluating the Effect of Open Student Models on Self-Assessment. Int. J. of Artificial Intelligence in Education 17(2), pp. 121-144.

[105] Morbidoni, C., Polleres, A., Tummarello, G., and Le-Phuoc, D. (2007). SemanticWeb Pipes. Technical Report. Available online at http://pipes.deri.org/

[106] Moulaison, H.L. (2008). Social tagging in the web 2.0 environment: author vs. user tagging. Journal of Library Metadata 8 (2), pp. 101–111.

[107] http://www.nextgenlearning.org/the-program/ Accessed online last time on 2012-07-12

[108] Pekczynski, P., Steiner, C., Mazzetti, A., and Barak, N. (2011) Second empirical evaluation report in corporate settings. Grapple D10.5 Deliverable, Version, 1.

[109] Phuoc, D.L., Polleres, A., Tummarello, G., Morbidoni, C. (2008). DERI Pipes: visual tool for wiring Web data sources.

[110] Quintarelli, E (2005). Folksonomies: power to the people. June 2005. http://www-dimat.unipv.it/biblio/isko/doc/folksonomies.htm.

[111] Romero, C. (2011). Handbook of educational data mining. Chapman & Hall/CRC - data mining and knowledge discovery series, 2011, Boca Raton, FL.

[112] Romero, M., Tricot, A., and Mariné, C. (2009). Effects of a context awareness tool on students' cognition of their team-mates learning time in a distance learning project activity. In Proceedings of the 9th international conference on CSCL - Volume 1 pgg. 641 - 645. Rhodes, Greece

[113] Romero, C., Ventura, S. (2010). Educational Data Mining: A Review of the State-of-the-Art. IEEE Transaction on Systems, Man, and Cybernetics, Part C: Applications and Reviews. 40(6), pp. 601-618.

[114] Rovai, A. (2002). Building Sense of Community at a Distance. The International Review Of Research In Open And Distance Learning, 3(1).

[115] Russell, S. J. and Norvig, P. (2003). Artificial Intelligence: A Modern Approach (2nd ed.)", Upper Saddle River, NJ: Prentice Hall, ISBN 0-13-790395-2.

[116] Scaife, M., Rogers, Y. (1996). External cognition: how do graphical representations work? International Journal of Human-Computer Studies, Volume 45, Issue 2, August 1996, Pages 185-213, ISSN 1071-5819, 10.1006/ijhc.1996.0048. Available online at http://www.sciencedirect.com/science/article/pii/S1071581996900488

[117] Schutz, D. (2009). Educational Intelligence: Bringing Schools into the 21st Century. Available at SSRN: http://ssrn.com/abstract=1336243

[118] Shahrour, G., Bull, S. (2008). Does Notice Prompt Noticing? Raising Awareness in Language Learning with an Open Learner Model. In: AH2008 Springer-Verlag, pgg. 173 - 182.

[119] Shankar, A., Louis, S. J., Dascalu, S., Hayes, L. J., Houmanfar, R. (2007). User-context for adaptive user interfaces. In Proceedings of the 12th international Conference on intelligent User interfaces (Honolulu, Hawaii, USA, January 28 - 31, 2007). IUI '07. ACM, New York, NY, 321-324

[120] Shneiderman, B. and Plaisant, C. (2004). Designing the User Interface: Strategies for Effective Human-Computer Interaction. 4th ed. Addison Wesley, 2004 ISBN 0-321-19786-0

[121] Shneiderman, B., Plaisant, C. (2005). Designing the User Interface. Chapter 14.5: Information Visualization (pp. 580-603). Boston: Pearson

[122] Spence, R. (2007). Information Visualization: design for interaction, 2nd ed. Pearson Education/Prentice Hall.

[123] Steiner, C., and Hillemann, E. (2010). Refinement and Improvement of Evaluation Framework. Grapple D8.1b Deliverable, Version, 2.

[124] Suh, B., Chi, E., Kittur, A., and Pendleton, B.A. (2008). Lifting the veil: improving accountability and social transparency in Wikipedia with wikidashboard. In Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems (CHI '08). ACM, New York, NY, USA, 1037-1040. DOI=10.1145/1357054.1357214 http://doi.acm.org/10.1145/1357054.1357214

[125] Tang, T.Y., McCalla, G. (2004). Evaluating a Smart Recommender for an Evolving E-learning System: A Simulation-Based Study. Canadian Conference on AI 2004. pp. 439-443

[126] Tanimoto, S. (2005). Dimensions of Transparency in Open Learner Models. Proc Int. Workshop on Learner Modelling for Reflection, to Supporter Learner Control, Metacognition and Improved Communication between Teachers and Learners, AIED, IOS Press, pgg. 100 - 106. Amsterdam, 2005.

[127] M. Tschannen-Moran, M., Wolfolk Hoy, A. (2001). Teacher efficacy. Capturing an elusive construct. Teaching and Teacher Education, vol. 17, pp. 783-805.

[128] Watkins, R., Leigh, D., Foshay, R. and Kaufman, R. (1998). Kirkpatrick Plus: Evaluation and Continuous Improvement with a Community Focus. Educational Technology Research & Development, 46(4): 90-96.

[129] Van Der Sluijs, K., Houben, G., (2006). A generic component for exchanging user models between web-based systems. International Journal of Continuing Engineering Education and Life-Long Learning 2006 - Vol. 16, No.1/2 pp. 64 - 76

[130] Van Der Sluijs, K., and Hover, K. (2009). Integrating Adaptive Functionality in a LMS. International Journal Of Emerging Technologies In Learning (IJET), 4(4). Retrieved December 31, 2009, from http://online-journals.org/i-jet/article/view/958

[131] Van Labeke, N., Brna, P., Morales, R. (2007). Opening up the Interpretation Process in an Open Learner Model, International Journal of Artificial Intelligence in Education 17(3), pp. 305-338.

[132] Vassileva, J., Sun, L. (2007). Using Community Visualization to Stimulate Participation in Online Communities.e-Service Journal, 6 (1), pp. 3-40.

[133] Vatrapu, R., Teplovs, C., Fujita, N., Bull, S. (2011). Towards visual analytics for teachers' dynamic diagnostic pedagogical decision-making. In Proceedings of the 1st International Conference on Learning Analytics and Knowledge (LAK '11). ACM, New York, NY, USA, pgg. 93-98. http://doi.acm.org/10.1145/2090116.2090129

[134] Yudelson, M. and Brusilovsky, P. (2005). NavEx: Providing navigation support for adaptive browsing of annotated code examples. In: C.-K. Looi, G. McCalla, B. Bredeweg and J. Breuker (Eds.). Proceedings of 12th International Conference on Artificial Intelligence in Education, AI-Ed'2005, Amsterdam, July 18-22, 2005, IOS Press, pp. 710-717.

[135] Zapata-Rivera, J.D., Greer, J.E. (2004). Visualizing and inspecting Bayesian belief models. In: International Journal of Artificial Intelligence in Education IJCAI, 14, pp. 1-37. IOS Press.

[136] Zhang, H., Almeroth, K. (2010). Moodog: Tracking Student Activity in Online Course Management System. Journal of Interactive Learning Research, 21(3), 407-429.