

Towards portable learning analytics dashboards

Andrii Vozniuk
EPFL
Lausanne, Switzerland
andrii.vozniuk@epfl.ch

Sten Govaerts
EPFL
Lausanne, Switzerland
sten.govaerts@epfl.ch

Denis Gillet
EPFL
Lausanne, Switzerland
denis.gillet@epfl.ch

Abstract—This paper proposes a novel approach to build and deploy learning analytics dashboards in multiple learning environments. Existing learning dashboards are barely portable: once deployed on a learning platform, it requires considerable effort to deploy the dashboard elsewhere. We suggest constructing dashboards from lightweight web applications, namely widgets. Our approach allows to port dashboards with no additional cost between learning environments that implement open specifications (OpenSocial and ActivityStreams) for data access and use widget APIs. We propose to facilitate reuse by sharing the dashboards and widgets via a centralized analytics repository.

Index Terms—learning analytics; dashboards; portability; open standards; widget; ActivityStreams; OpenSocial.

I. INTRODUCTION

Nowadays we observe a change in the way we learn, as education becomes more flexible and distributed [1], [2]. Participating in courses remotely is a common practice, whether in an institution or using massive open online courses (MOOCs). The distributed and flexible nature of the learning process creates new challenges. It becomes much harder for teachers to observe, control and adjust the learning process. For instance, in the MOOCs, where thousands of students are simultaneously taking an online class, it is impossible for a teacher to consider individual capabilities and preferences of each learner. On the other hand, when learners study remotely, they tend to get less involved in the process and eventually loose motivation. Consequently, MOOCs experience a much higher drop out rate, compared to ordinary classes [3].

Learning analytics (LA) aims to address such issues by collecting and using traces that learners leave online [4]. According to Siemens et al. [5], LA is “the measurement, collection, analysis and reporting of data about learners and their contexts, for purposes of understanding and optimizing learning and the environments in which it occurs”. Teachers and learners can benefit differently from LA. A teacher could observe live statistics about what and how learners are doing and based on this intervene and adapt the course when required. For example, a LA-based notification system can predict which students are about to drop out [6]. Then again, learners’ motivation might improve if students are able to compare their individual progress with their peers. In general, LA aim to help us to better understand how we learn and improve our learning [4].

Various online learning management systems (LMSs) and personalized learning environments (PLEs) [7] exist that support learning at distance. Examples include Blackboard [8],

Moodle [9] and Graasp [10]. Such systems often lack or provide limited LA dashboards. A dashboard is a visual display of the most important information needed to achieve one or more learning objectives, consolidated and arranged on a single screen so the information can be monitored at a glance [11]. Duval et al. propose to use LA dashboards for representing important information about learners [4]. Support for such dashboards could, for instance, increase the use of LMSs and PLEs for MOOCs to provide an overview to learners and teachers.

Through the use of plugins, learning dashboards are often integrated into learning environments that lack LA. Moodle is such a plugin [12] for Moodle and other LA plugins exist to track learners using blogs [13]. In addition, general-purpose analytics tools are often used, although they are not tailored towards learning. For instance, Google Analytics [14] is used to observe resource usage by students. Using such external services hosted by third parties for tracking students causes privacy concerns. Most existing LA solutions are either not specifically designed for learning (e.g. Google Analytics) or are tightly coupled to a learning platform (e.g. Moodog). It usually requires a considerable effort, most often involving programming, to adapt a learning dashboard or deploy it on another platform. This limits the adoption of LA tools.

In this paper we propose an alternative solution: the use of portable widgets and open specifications to build portable learning dashboards. Our approach enables easy deployment, customization, extension and reuse of learning dashboards. The paper is structured as follows: in section 2 we define prerequisites for portable analytics and discuss why existing solutions are not portable; section 3 presents how open standards can be used for capturing and storing learning data; section 4 describes methods to build learning dashboards based on portable widgets; section 5 discusses the architecture; section 6 presents on-going integration of our approach into Graasp and section 7 concludes this paper.

II. PORTABILITY REQUIREMENTS

To understand the concept of portable learning dashboards, we first define software portability in general. Software portability is the ability to run the same software on different platforms with no or little effort [15]. Some configuration of LA tools on a new platform might still be required but its cost is considerably lower than developing the tools from scratch.

A LA component interacts with a learning platform through a collection of interfaces. To facilitate portability, all interfaces for a LA component have to be made identical across platforms [15]. Ideally, we would like such common interfaces to be already implemented by all learning platforms. Otherwise, adaptation of the interfaces or dashboards will be required during the porting process [15].

We have identified three important classes of LA solutions based on their integration with a learning platform. For each case we discuss the portability.

Tightly coupled LA solutions are developed for a particular platform (e.g. Blackboard [8]). Such solutions use Application Programming Interfaces (APIs) specific to the platform to access the data and commonly the user interface (UI) for such LA is integrated in the learning platform. Consequently, these solutions do not work on other platforms.

Pluggable LA solutions are integrated in a platform via a plugin interface (e.g. in Moodle). Such a plugin interface is often not compatible across platforms. If a user switches to another platform then she needs a LA plugin for that platform.

Standalone LA solutions can be divided into two subcategories: (i) general web analytics services (e.g. Google Analytics¹ and Woopra²) and (ii) services developed specifically for learning analytics (e.g. the CAM web service [16]).

General web analytics services can be regarded as portable, although they have other limitations. The main limitation of web analytics services is that because of their general scope, domain-specific information is not taken into account. For instance, just tracking that a learner visited a page, is not sufficient to deduce the activity she was involved in, e.g. reading, watching a video or playing a game. Such web analytics tools are not able to track information specific to learning. Another shortcoming, essential for many educational institutions, is that privacy-sensitive tracking data of learners are stored on third-party servers. Due to privacy policies and laws, data of learners can often not leave the institution.

Standalone LA solutions developed specifically with LA in mind also face some issues. Contextualized Attention Metadata (CAM) aims to capture contextual information regarding the activity of a user [17]. CAM is very flexible in what it can track, which also limits its portability. CAM has no common vocabulary to describe actions and allows each platform to define its custom vocabulary, which makes it difficult to develop fully portable dashboards on top of the CAM specification. Besides, the CAM specification has not been widely adopted outside research so far.

Our goal is to achieve portability by implementing learning dashboards as external pluggable components and using well-defined interfaces with a learning platform. The specification of these interfaces should be based on existing open standards. In this way we can ensure that our solution is license-free and extendable. In addition, using open standards increases compatibility with existing and future platforms and tools,

because open standards often have a large user base that enables wider adoption and future support and extensions.

III. OPEN SPECIFICATIONS FOR DATA ACCESS

In order for LA tools to function, a learning platform must provide access to its stored data. According to object-oriented modeling [18], such data represents dynamic (behavioral) and static (structural) views of the platform [18]. The dynamic view focuses on the dynamic behavior of the learning platform based on the collaboration between users, tools and learning resources and changes of their internal states. On the other hand, the structural view describes the static structure of the learning platform using the users, tools and learning resources and their relations [18]. LA tools often exploit both views.

In the remainder of this section, we elaborate on how these two views can be represented for LA. The first subsection focuses on the dynamic view to represent user activities. Afterwards, we expand on how to model structural social data. Finally we discuss APIs to access these two types of data.

A. *ActivityStreams* to represent user activities

In essence, the dynamic view of a system in object-oriented modeling [18] emphasizes the interactions between users, resources and tools. Storing data about user actions is a common problem for various social platforms (including learning environments). Such platforms track user actions for diverse reasons, e.g. (i) to increase user awareness of their activities and those of peers on the platform, or (ii) to enable personalization techniques (e.g. personalize search result ranking or advertisements). To make such techniques even more powerful, user actions on multiple platforms could be aggregated to provide a full picture of the user's activities. To achieve this, all platforms should exchange user action data.

Currently, many social media platforms provide a data feed of activities of the user and her social network in RSS format³. The RSS format was developed to publish frequently updated media, such as blog entries, comments or videos. As such it is not suitable for capturing all rich metadata related to user activities. The ActivityStreams specification⁴ is designed specifically to model user interactions and enables platforms to share detailed information on user activities.

An activity stream is a sequence of actions made by a user. Such a stream in the ActivityStreams specification models the story of a person performing an action on or with an object. Technically in the ActivityStreams specification, an action is represented as a 5-tuple: (*Published*, *Actor*, *Verb*, *Object*, *Target*). *Published* represents the time at which the activity was published. *Actor* defines the actor that performs the action. *Verb* describes which action is performed. The specification defines a extendable set of verbs. *Target* is intended to describe the consequences of the action. Consider this example action: "Alice added a book by Tom Sawyer to her 'Favorite Books' collection". In this example, "Alice" is an *Actor*, "added" a

¹<http://www.google.com/analytics/>

²<http://www.woopra.com/>

³For instance, every Facebook page has an associated RSS feed, see <http://ahrengot.com/tutorials/facebook-rss-feed/>

⁴<http://activitystrea.ms/specs/json/1.0/>

Verb, “a book by Tom Sawyer” an *Object* and “‘Favorite Books’ collection” is a *Target* of the action.

ActivityStreams has a large uptake and is supported by most social media platforms (e.g. Google+⁵ and Facebook⁶). Many organizations have contributed to the development of the specification⁷ and made it an open standard through the Open Web Foundation Final Specification Agreement⁸ that allows the use, extension and sale to anyone.

Due to the broad definition and extensibility⁹ of the ActivityStreams specification, it can be applied to the learning domain [19]. ActivityStreams can be used to represent interactions of learners and teachers with the learning platform and is thus an appropriate specification to represent activity data for LA. Man et. al. also propose ActivityStreams as a standard for exchanging user activities among learning platforms [19].

Various specifications to capture interactions have been developed [20]. One of these, used for educational purposes, is CAM [17]. In contrast to CAM, Activity Streams has a defined set of verbs with a corresponding meaning to describe actions¹⁰. This verb definition enables better interoperability across platforms, which leads to better portability.

B. OpenSocial for the structural view

The large uptake of social media platforms has fostered the adoption of social media features in learning environments, e.g. social networks for collaboration and blogs are integrated in most LMSs. Such social features provide a rich data source for LA (e.g. for social network analysis). Hence, in addition to the user activity data, information on the users, resources, tools and their relationships is valuable for LA. Such data are described by the structural view in object-oriented modeling [18].

The OpenSocial specification¹¹ aims to provide a common method for storing and accessing social data. Additionally, OpenSocial defines a component-hosting environment (a container) and a set of common APIs for web applications. The standardization of the interfaces to store and retrieve data enables the development of components, which are portable across systems. At the moment, OpenSocial is adopted by many popular social media platforms, e.g. Ning¹², MySpace and Orkut¹³. Originally OpenSocial was focused on social network applications, but thanks to educational interest in social media, OpenSocial became more popular in technology enhanced learning (TEL). Notable TEL examples include Graasp¹⁴, Sciverse¹⁵ and ROLE [21].

⁵<https://developers.google.com/+api/latest/activities>

⁶<http://developers.facebook.com/blog/post/225/>

⁷<http://wiki.activitystrea.ms/w/page/24500522/Implementors>

⁸<http://openwebfoundation.org/legal/the-owf-1-0-agreements/owfa-1-0>

⁹<http://activitystrea.ms/specs/json/1.0/>

¹⁰<http://activitystrea.ms/specs/json/schema/activity-schema.html>

¹¹<http://docs.opensocial.org/display/OSD/Specs>

¹²developer.ning.com/docs/opensocial/1.0/index.html

¹³<https://developers.google.com/orkut/articles/tutorial/tutorial>

¹⁴<http://graasp.epfl.ch/>

¹⁵<http://surfnet.nl/en/nieuws/Pages/ConstructionstartsofOpenSocialenvironment>

The OpenSocial 2.0 specification incorporates an API for accessing user activities using the ActivityStreams format. In this manner, OpenSocial 2.0 combines both the dynamic and structural views of a social or learning platform.

IV. ASSEMBLING LEARNING DASHBOARDS USING WIDGETS

In the previous section we have discussed the interface specifications to exchange data between platforms and applications. To achieve our goal of creating portable LA dashboards, it should be possible to simply take LA tools from one platform and run them on another platform. To ease such portability a specification to deploy LA dashboards is required. As discussed in the previous section, we are interested in tools that use ActivityStreams and OpenSocial APIs. To this end OpenSocial widgets fit this requirement, as they are small lightweight web applications with excellent integration with the OpenSocial specification. OpenSocial widgets can run immediately on a platform implementing the OpenSocial API.

Using widgets for LA dashboards has several benefits. First, dashboards can be constructed using diverse widgets that display visualisations and metrics. These widgets can be arranged on a single screen to provide all information at a glance [11]. Second, as OpenSocial widgets are programmed using just Javascript and HTML, the widgets simply run in a browser, which provides an operating system-independent solution. Javascript and HTML have good support on mobile devices as no additional plugins are required in contrast to Flash or Java solutions. Additionally, the platform can also control access to the data that is exposed through the OpenSocial APIs with privacy policies. For instance, teachers can observe trends over the whole class, while students might only see and reflect on their own actions.

When we organize several widgets as a dashboard, such an assembled dashboard could be packaged as a widget in itself. In this case a dashboard is a widget containing other widgets or also called a metawidget [22]. If a dashboard is bundled as a metawidget, the whole assembled set of widgets become portable. Metawidgets allow teachers and learners to share assembled LA dashboards across platforms.

V. ARCHITECTURE

In summary, to achieve portable LA dashboards we propose to build dashboards with OpenSocial widgets that use the OpenSocial API to retrieve data for visualization and analysis. User activity data, which is essential for LA, is represented using the ActivityStream specification. We propose to distribute the learning dashboard assembled with several widgets as a metawidget itself. For easier distribution, LA dashboards and widgets can be added to a widget repository where teachers and learners can find and easily add them to their learning environment.

The proposed architecture (see Figure 1) is composed of:

- 1) *Presentation tier* consists of the learning platform UI and a learning dashboard, constructed from widgets. An example of such a UI is shown in Figure 2.

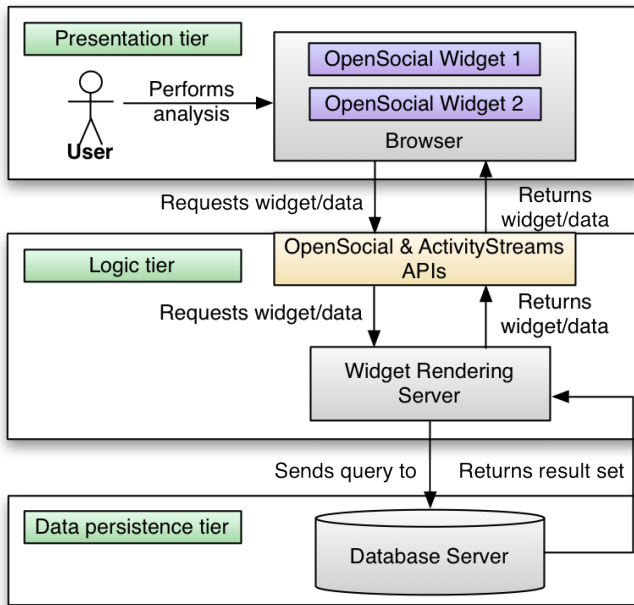


Fig. 1. The presentation tier obtains data from the logical tier via OpenSocial and ActivityStreams APIs.

- 2) *Logic layer* provides access to the ActivityStreams and OpenSocial APIs that are used to communicate with presentation and logic tier.
- 3) *Data persistence tier* stores the activity and social data.

VI. PORTABLE LEARNING ANALYTICS IN GRAASP

To demonstrate our approach, we are currently integrating the proposed architecture into the Graasp platform [23] to enable portable LA dashboards. Graasp supports people in their personal and collaborative activities. This can be any kind of activity, but Graasp is often used as a PLE. Users can organize their activities into public or private contextual spaces [10], where they can share relevant resources and tools with other users. Unlike mainstream social media platforms, Graasp focuses on activity-based relations (e.g. people studying English), instead of individual social networks (e.g. people who I know). In the remainder of this section we discuss how this on-going development will be achieved.

Graasp is built as a single page JavaScript application that supports OpenSocial widgets through the Apache Shindig container¹⁶. Shindig is a free open-source widget container with a reference implementation of the OpenSocial and ActivityStreams specification. By integrating Shindig into the Graasp platform, widgets have read and write access to Graasp data through the OpenSocial APIs (e.g. user information or activity streams). This enables the creation of LA dashboards, since widgets can access user activities and social data for analysis or visualization through these APIs. Figure 2 shows a prototype LA dashboard running on Graasp. Such LA dashboards assembled with OpenSocial widgets can run on

any platform that implements the OpenSocial and ActivityStreams specifications. To ease distribution and search of LA dashboards and widgets Graasp has integrated the ROLE Widget Store [21] as a widget repository.

Through the Space extension¹⁷ of OpenSocial that is implemented in Graasp, LA can be made context-aware. A space is a place where a group of people can collaborate on an activity. The resources and widgets used in such a space are shared among the people in the space [24] (see Figure 2). By being context-aware (i.e. related activities are carried out in a given online workplace), roles and privacy settings can be enforced that enable control over the selected LA data in the dashboard. Thus a space can provide LA for a specific context. For instance, in a space for learning English only the data of English learners would be shown in the LA dashboard. Through spaces, we can track learners and their context and we can provide detailed LA tools for that context only.

Context can also be captured while tracking user activities. Currently Graasp tracks the following activities related to users, resources, applications and spaces (in accordance with ActivityStreams Specification): *add*, *update*, *remove*, *invite*, *join*, *leave*, *favorite* and *tag*. The ActivityStreams specification supports contextual information through the *Target* field (see Section III-A). In our example, it is not only important that Alice read a book, but the context is also valuable, e.g. while learning English. This could be tracked as: “Alice read a book by Tom Sawyer in the ‘Learning English’ space”.

VII. CONCLUSION

This paper discusses issues related to the portability of learning analytics solutions. We propose to enable portability of LA dashboards by building dashboards based on widgets that implement the OpenSocial and ActivityStreams specifications. Additionally, we propose to construct LA dashboards as metawidgets using a set of widgets as building blocks. This modular approach encourages customization, as the dashboard consists of interchangeable widgets that can be adapted and extended easily. The metawidget approach allows sharing of assembled dashboards between users and across platforms. In order to foster sharing and easy deployment, we propose to establish and use a repository of LA widgets.

By providing a portable learning analytics dashboards, teachers and learners participating in distance education or MOOCs can have access to LA tools to support their learning and increase their awareness on personal and group activities. The portability of the LA tools allows a more free choice of learning environment and the modular widget-approach to create LA dashboards allows users to share and re-use dashboards in various scenarios and platforms.

We are currently in the process of integrating the proposed architecture into Graasp and validating it with science teachers in secondary schools. In conclusion, we would like to stress that our approach is more broadly applicable than just for LA. It can be employed for general user activity analysis (e.g.

¹⁶<http://shindig.apache.org/>

¹⁷<http://docs.opensocial.org/display/OSD/Space+Proposal>

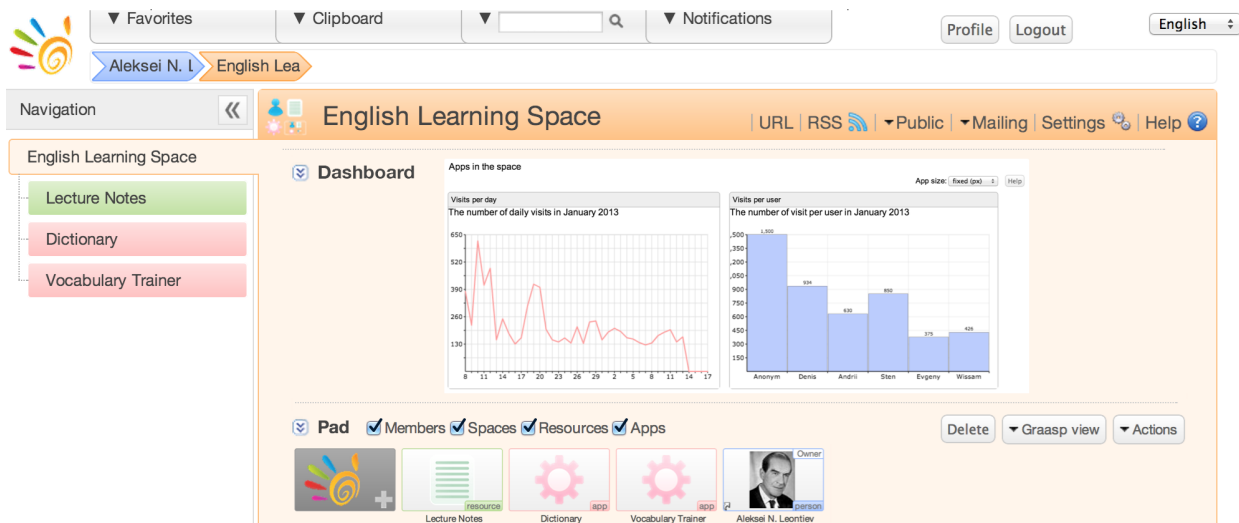


Fig. 2. A sample learning dashboard in Graasp.

software development analysis) or even more general tools that make use of user activities (e.g. recommendation services).

ACKNOWLEDGMENT

We would like to thank Sergii Vozniuk for the fruitful discussions and comments. This research is partially funded by the PLE Project of the Swiss AAA/SWITCH Program and the European Union in the context of the Go-Lab project (Grant Agreement no. 317601) under the Information and Communication Technologies theme of the 7th Framework Programme for R&D.

REFERENCES

- [1] D. Gillet, "Tackling engineering education research challenges: Web 2.0 social software for personal learning," *International Journal of Engineering Education*, vol. 26, no. 5, p. 1134, 2010.
- [2] D. Gillet, A. Ngoc, and Y. Rekik, "Collaborative web-based experimentation in flexible engineering education," *IEEE Transactions on Education*, vol. 48, no. 4, pp. 696–704, 2005.
- [3] A. Koutropoulos, M. Gallagher, S. Abajian, I. de Waard, R. Hogue, N. Keskin, and C. Rodriguez, "Emotive vocabulary in MOOCs: Context & participant retention," *The European Journal of Open, Distance and E-Learning*, vol. 1, 2012.
- [4] E. Duval, "Attention please! learning analytics for visualization and recommendation," in *Proceedings of LAK11: 1st International Conference on Learning Analytics and Knowledge*, 2011, pp. 9–17.
- [5] G. Siemens, D. Gasevic, C. Haythornthwaite, S. Dawson, S. Shum, R. Ferguson, E. Duval, K. Verbert, and R. Baker, "Open learning analytics: an integrated & modularized platform," *The Society for Learning Analytics Research*, 2011.
- [6] J. Heppen and S. Therriault, *Developing early warning systems to identify potential high school dropouts*. American Institutes for Research, National High School Center, 2008.
- [7] D. Gillet, E. Law, and A. Chatterjee, "Personal learning environments in a global higher engineering education web 2.0 realm," in *Education Engineering (EDUCON)*. IEEE, 2010, pp. 897–906.
- [8] P. Bradford, M. Porciello, N. Balkon, and D. Backus, "The Blackboard learning system: The be all and end all in educational instruction?" *Journ. of Edu. Technology Systems*, vol. 35, no. 3, pp. 301–314, 2007.
- [9] M. Dougiamas and P. Taylor, "Moodle: Using learning communities to create an open source course management system," in *World conference on educational multimedia, hypermedia and telecommunications*, vol. 2003, no. 1, 2003, pp. 171–178.

- [10] E. Bogdanov, S. El Helou, D. Gillet, C. Salzmman, and S. Sire, "Graasp: a web 2.0 research platform for contextual recommendation with aggregated data," in *Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems*. ACM, 2010, pp. 3523–3528.
- [11] S. Few, "Dashboard confusion revisited," *Perceptual Edge*, 2007.
- [12] H. Zhang, K. Almeroth, A. Knight, M. Bulger, and R. Mayer, "Moodog: Tracking students' online learning activities," in *World Conference on Educational Multimedia, Hypermedia and Telecommunications*, vol. 2007, no. 1, 2007, pp. 4415–4422.
- [13] R. Ferguson, S. Buckingham Shum, and R. Deakin Crick, "Enquiry-blogger: using widgets to support awareness and reflection in a PLE setting," 2011.
- [14] P. Betty, "Assessing homegrown library collections: Using Google Analytics to track use of screencasts and flash-based learning objects," *Jour. of Electr. Resources Librarianship*, vol. 21, no. 1, pp. 75–92, 2009.
- [15] J. Mooney, "Bringing portability to the software process," *Dept. of Statistics and Comp. Sci., West Virginia Univ., Morgantown WV*, 1997.
- [16] J. Najjar, E. Duval, and M. Wolpers, "Attention metadata: Collection and management," in *WWW2006 workshop on logging traces of web activity: the mechanics of data collection*, 2006, pp. 1–4.
- [17] M. Wolpers, J. Najjar, K. Verbert, and E. Duval, "Tracking actual usage: the attention metadata approach," *Journal Of Educational Technology And Society*, vol. 10, no. 3, p. 106, 2007.
- [18] J. Holt, *UML for Systems Engineering: watching the wheels*. The Institution of Engineering and Technology, 2004.
- [19] H. Man, H. Chen, and Q. Jin, "Open learning: A framework for sharable learning activities," *Advances in Web-Based Learning-ICWL 2010*, pp. 387–392, 2010.
- [20] V. Butoianu, P. Vidal, K. Verbert, E. Duval, and J. Broisin, "User context and personalized learning: a federation of contextualized attention metadata," *Journal of Universal Computer Science*, vol. 16, no. 16, pp. 2252–2271, 2010.
- [21] S. Govaerts et al., "Towards responsive open learning environments: the role interoperability framework," in *Proc. of the 6th Euro. conf. on Technology enhanced learning, EC-TEL'11*. Springer-Verlag, 2011, pp. 125–138.
- [22] M. Blattner, E. Glinert, J. Jorge, and G. Ormsby, "Metawidgets: towards a theory of multimodal interface design," in *Computer Software and Applications Conference, 1992. COMPSAC'92. Proceedings., Sixteenth Annual International*. IEEE, 1992, pp. 115–120.
- [23] E. Bogdanov, F. Limpens, N. Li, S. El Helou, C. Salzmman, and D. Gillet, "A social media platform in higher education," in *Global Engineering Education Conference (EDUCON), 2012 IEEE*. IEEE, 2012, pp. 1–8.
- [24] E. Bogdanov, C. Salzmman, and D. Gillet, "Contextual spaces with functional skins as opensocial extension," in *ACHI 2011, The Fourth International Conference on Advances in Computer-Human Interactions*, 2011, pp. 158–163.