Cloud-Savvy Contextual Spaces as Agile Personal Learning Environments or Informal Knowledge Management Solutions

Denis Gillet, Evgeny Bogdanov

▶ **To cite this version:**

Denis Gillet, Evgeny Bogdanov. Cloud-Savvy Contextual Spaces as Agile Personal Learning Environments or Informal Knowledge Management Solutions. Proceedings of the 12th International Conference on Information Technology Based Higher Education and Training, Oct 2013, Antaliya, Turkey. hal-00979785

Cloud-Savvy Contextual Spaces as Agile Personal Learning Environments or Informal Knowledge Management Solutions

Denis Gillet and Evgeny Bogdanov
School of Engineering
École Polytechnique Fédérale de Lausanne (EPFL)
Lausanne, Switzerland
denis.gillet@epfl.ch and evgeny.bogdanov@epfl.ch

Abstract—**This paper presents how advanced social media platforms can be exploited to construct and share contextual spaces enabling the instantiation of agile personal learning environments or informal knowledge management solutions. The usefulness of open plugins to collect resources from the cloud in such dedicated contextual spaces is discussed. The mechanisms for the personalization of spaces from an interaction point of view once populated with resources and their sharing across platforms are also detailed. Results are illustrated with the case of the *Graasp* platform developed in the framework of European research projects in Technology Enhanced Learning.**

Keywords—Personal Learning Environments; Knowledge Management; Creative Commons; Online Shared Spaces; Cloud

I. INTRODUCTION

A personal learning environment (PLE) is a recent and evolving concept yearly discussed in the framework of the PLE conference[1] and in workshops related to Technology Enhanced Learning (TEL). Informal knowledge management solutions are more and more required in flat organizations such as non-governmental organizations (NGO) in which learning is an informal process supported through practice sharing in the field. This paper presents how advanced social media platforms can be exploited to construct and share contextual spaces enabling the instantiation of personal learning environments or informal knowledge management solutions (KMS). The usefulness of open plugins to collect resources from the cloud in such dedicated contextual spaces is discussed. The mechanisms for the personalization of spaces from an interaction point of view once populated with resources and their sharing across platforms are detailed. Results are illustrated with the case of the *Graasp* platform.

The rest of the paper is organized as follows. Section II detailed the concept of online spaces as shared contexts for the aggregation of cloud resources and services, as well as its formalization as part of the OpenSocial specification. Section III describes the *Graasp* social media platform developed to enable the creation and the exploitation of online spaces for collaborative learning and knowledge management. Section

IV introduces the plugin architecture devised to enable a one-click aggregation of cloud resources in OpenSocial spaces hosted in *Graasp*. Section V tackles the question of space personalization enabled through the creation of user-defined skins providing alternative interaction features and visualization schemes. Section VI illustrates the benefits of the OpenSocial space specification exploited in conjunction with the OMDL standard to enable the exchange of application bundles between social media platforms and repositories. Section VII emphases the benefit of pushing the usage of Creative Commons licenses to ease the aggregation, the sharing and the repurposing of cloud resources. Finally, Section VIII discuses the similarities existing in the usage of online spaces for the construction and the exploitation of both personal learning environments and knowledge management solutions. The paper ends with concluding remarks.

II. OPENSOCIAL SPACES

PLE enabling technologies are developed in several research projects and initiatives such as the ROLE European project[2], which focuses on defining and validating a sound pedagogical framework for self-directed and informal learning, as well as a sound technical framework for Open Web 2.0 PLE [1]. The investigations in ROLE rely on participatory design and social requirement engineering conducted in academic and professional test beds. Currently, the definition we propose for a PLE is *any open online environment or social media platform exploited by users through interactive devices for learning and knowledge management*. In this definition, *open* means coming from the cloud in contrast to proprietary institutional resources. *Devices* are mentioned to emphasize the ubiquitous and sometime tangible nature of the resources used to gather or access resources. In addition, when dealing with knowledge artifacts and online communities, there is a blurring distinction between their exploitation for *knowledge management* or *learning*, as the latter often occurs without noticing when practicing the former. The following properties of a PLE can be listed: i) Any digital ecosystem repurposed for learning is a PLE, i.e., it is the intention of use and not the design of the

---

[1] http://pleconf.org

[2] http://www.role-project.eu

platform which makes it a PLE [2]; ii) Constructing the environment is part of the learning process, i.e. constructionism is finally extended at the level of the environment; iii) PLE are personal but not individual, they may integrate peers, coaches, teachers or even relatives [3]; iv) Each PLE is designed for a single context or purpose. Hence, the corresponding space is referred to as a contextual space.

Participatory design activities carried out in recent years have demonstrated that, to be complete, a PLE or a KMS should integrate a contextualizing entity we define as a shared *activity space*, each space integrating *people* (members), *resources* (shared digital artifacts), *apps* (tools or end-user services offered as widgets or OpenSocial apps to enable the realizing of contextual actions or visualization), as well as subspaces. As such, a space is an instantiation by an individual of a PLE or a KMS constructed to support a dedicated activity [4]. As a space usually gathers from the cloud people (enabling interaction), resources (enabling knowledge acquisition and consolidation) and apps (enabling environment plasticity), it strongly relies on advanced *search*, *recommendation* and *aggregation* features.

In order to enforce the personal nature of a PLE and to enable its sharing indifferently of user preferences, a space should not be captive of a single platform. As a consequence, we are standardizing[3] this concept in the OpenSocial framework to enable integration and portability between exiting and future OpenSocial containers.

III. *GRAASP*: A SOCIAL MEDIA PLATFORM FOR COLLABORATIVE LEARNING ACROSS BOUNDARIES

Graasp is a social media platform developed through participatory design in the framework of the Palette and the ROLE European projects to enable the support of communities of practice and the creation of personal learning environments without any intervention from institutions or managers. *Graasp* enables the creation of online spaces shared between people belonging to different communities and networks. Embedded shared resources are gathered across institutional and corporate boundaries. Unlike dominant social media, *Graasp* enables a fine definition of the audience, as well as the associated rights and roles to ensure trust construction and privacy enforcement. In *Graasp*, people map their personal and shared projects, interests, and activities into public or private contextual spaces integrating invited members, relevant resources and necessary apps which can be tagged and rated. Any space or resource in *Graasp* integrates its own discussion thread to enable contextual interaction. In addition to an innovative relation-based recommendation engine, one of the core features of *Graasp* is the *GraaspIt!* bookmarklet which supports an easy aggregation of cloud resources as detailed in the next section.

IV. CLOUD AGGREGATION AND *GRAASPIT!*

To ease resource aggregation from the cloud, we propose a plugin-based cloud aggregation architecture, where users can add resources from different Web sites into their spaces in a

---

---

few clicks. The cloud aggregation architecture consists of two parts: a Web browser bookmarklet and a plugin manager (Fig. 1). When the user visits a Web page and clicks on the bookmarklet, the JavaScript code is injected into the page and executed. This code sends the page URL to the plugin manager. The plugin manager finds a plugin that corresponds to the URL and sends the plugin code back to the browser. Once the plugin code is loaded in the browser, it scrapes the page data (such as a name, a description, an embed tag, or some other information available) and sends this information to the user's Web platform. Then, the Web platform processes the information and creates either a new resource or a new app based on the received information.
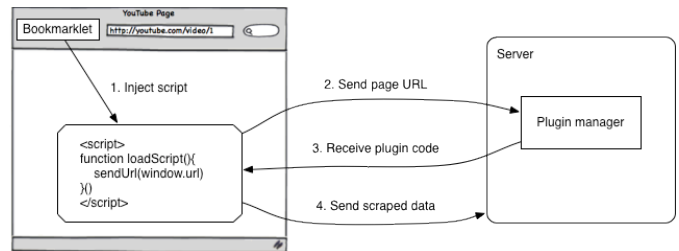

Fig. 1. Cloud aggregation architecture

In *Graasp*, the Cloud aggregation architecture is represented by the *GraaspIt!* bookmarklet and the corresponding plugin manager. *GraaspIt!* is a bookmarklet that can be activated from the browser bookmark bar at any time when surfing the Web (Fig. 2). Once a user is logged into *Graasp*, there is an option to add the bookmarklet to the browser bookmarks bar. Later, when the user visits a page of interest, the *GraaspIt!* button can be clicked and the user is provided with a dialog to add the item into *Graasp* (Fig. 3). After editing the name (if needed) and clicking on the "Add to Graasp" button, the resource will be immediately added into the personal *Graasp* clipboard. Then, it can be moved into the appropriate *Graasp* space with a simple drag-and-drop gesture. With *GraaspIt!* users can easily populate their spaces with resources and apps available in the Cloud.
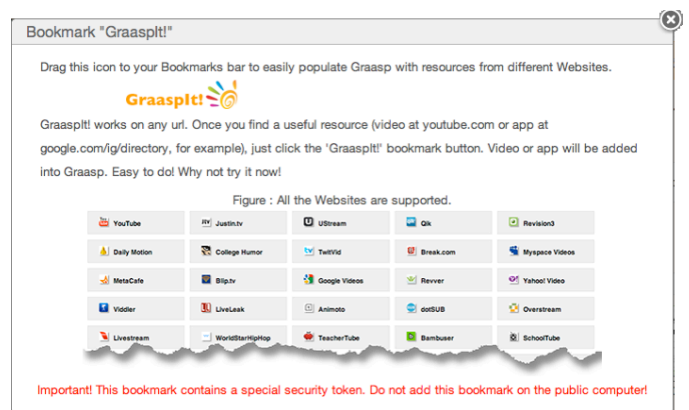

Fig. 2. The *GraaspIt!* feature integrated as a bookmark script enabling a one-click gathering of any online resource when surfing the Web
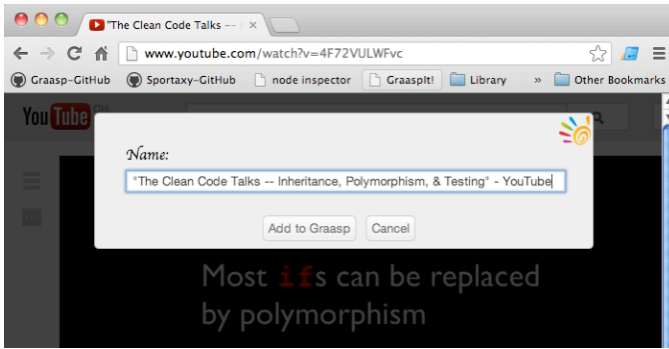
Fig. 3.   The *Graaspit!* dialog

GraaspIt! provides a unified way of aggregating information from various Web sources. It scrapes the information from the page the user opens in the browser and sends it to the *Graasp* backend. The *GraaspIt!* code is executed on the page when the user clicks on the bookmarklet. User identification is done via security token built into the bookmarklet. The bookmarklet sends a page URL to the plugin manager of *Graasp* as in the cloud aggregation architecture. In *Graasp*, the plugin manager can be further divided into three parts.

First, to extract the useful information from the page, the plugin manager tries to use the embed.ly library[4], which recognizes more than one hundred Web sites and enables the embedding of their content. If the action is successful and the corresponding embed.ly plugin found, then a new resource is created in *Graasp*. This resource is given the name, the description and the embed code obtained from the embed.ly service.

Second, if the first part of the plugin manager did not work (for example, the page is not supported by embed.ly), *GraaspIt!* tries to retrieve a customized plugin from *Graasp* for the selected Web site. Each JavaScript plugin is mapped to a Web site URL. Based on the mapping scheme, *GraaspIt!* selects which plugin should be used for the URL open in the browser. Afterwards, the JavaScript file with the relevant code is loaded and executed. For example, if a ROLE Widget Store[5] plugin is mapped to the URL *role-widgetstore.eu/tool*, then whenever the user clicks *GraaspIt!* when navigating the corresponding website, the ROLE Widget Store plugin will be loaded and executed. The plugin code extracts the name, the description and the embed code for the open URL according to the plugin logic and, then, *GraaspIt!* sends this information back to *Graasp* which creates a new resource or a new app based on the extracted data.

Third, if neither embed.ly works nor a customized plugin exists for the Web site, the third part of the plugin manager creates a screenshot of the Web page and saves it as a resource in *Graasp*, providing in such a way a combined bookmarking and archiving feature.

---

[4] http://embed.ly

[5] http://www.role-widgetstore.eu

The cloud aggregation architecture implemented in *Graasp* enables open content providers to add support for their own repositories or platforms via customized plugins. Content providers interested to create a *GraaspIt!* plugin for their Web platforms can use the existing *pluginmock.js* file as a basis for the development of their own plugins. This file can be found on Github[6]. As an example, let us consider the plugin that adds to *Graasp* in one-click OpenSocial apps available in the ROLE Widget Store (Fig. 4). The plugin file is called *widgetstore.js* and the matching URL for the service is *role-widgetstore.eu/tool*. Thus, when the user runs plugin on the pages matching this URL, the plugin *widgetstore.js* is called. As the code snippet in Fig. 4 shows, the plugin implements three scraping functions *findName*, *findEmbedCode* and *findThumbnail*, that return respectively the name, the URL and the thumbnail URL of the OpenSocial widget.

```
1  /**
2   * WidgetStore (widgetstore.js)
3   *
4   * matches urls: 'role-widgetstore.eu/tool'
5   *
6   */
7
8  ...
9
10  /**
11   * Returns the name of the OpenSocial app
12   */
13  findName: function () {
14    return $("#content-content .secondColumn .title")
15  },
16
17  /**
18   * Returns the thumbnail url of the OpenSocial app
19   */
20  findThumbnail: function () {
21    return $(".thumbnail-area img.imagecache-thumbnail").attr('src')
22  },
23
24  /**
25   * Returns the OpenSocial app url
26   */
27  findEmbedCode: function () {
28    return $('#edit-source-2-wrapper input').val()
29  }
```

Fig. 4.   The *GraaspIt!* plugin for ROLE Widget Store

V.   SPACE PERSONALIZATION

Once a space is created in *Graasp*, the core part of the interface (the Pad) enabling authorized users to interact with the embedded resources can be further personalized. In addition to the standard view (called the *Graasp* view) provided to populate spaces and to visualize their full content and members (Fig. 9), an extension mechanism is introduced to adapt the interaction and visualization mode through functional skins [5].

A functional skin is an XML file with some JavaScript code. It can be created by any user and added in a space at runtime without the intervention of developers. The functional skin feature can be seen as a client-side plugin. *Graasp* offers two built-in functional skins that can be selected using a popup menu: The Resource view (Fig. 5) and the App view (Fig. 6). The Resource view displays the list of all resources existing in a space and provides links for individual or full download. In addition, previews of resources can be displayed. The App view displays and activates all app instances from a space as a visual mashup. In this view, apps can be resized and

---

[6] https://github.com/react-epfl/graaspit

their order can be modified through drag and drop. The layout and the order of apps are persistent.
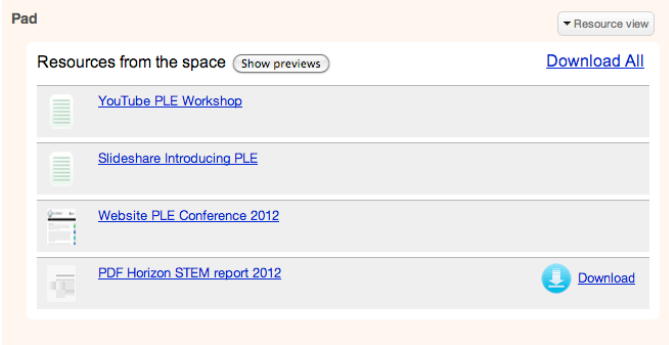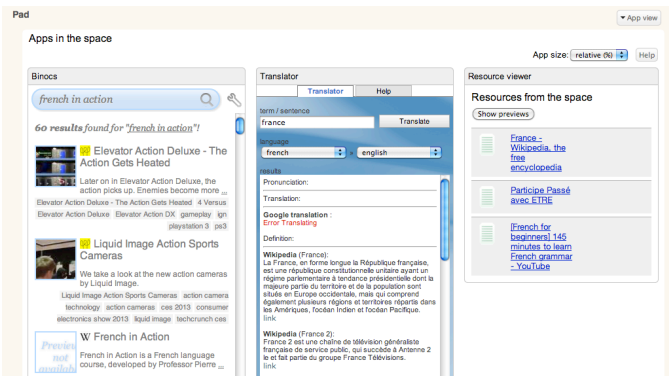

Fig. 5. Resource view for a space

Fig. 6. App view for a space

These two cases illustrate the fact that both the interaction features (as example letting users download a resource) and the visualization schemes (letting users seeing resources as a list or an active app) can be defined through skins.

VI. SPACE SHARING IN THE CLOUD

To support the interaction of people with spaces in the Cloud and enable sharing of created contexts with peers, *Graasp* implements several mechanisms to share and exchange app bundles (sets of apps combined together for a specific purpose) and spaces.

A bundle can be extracted from the existing space and exported as an OMDL file[7]. The OMDL file can be imported into another platform or shared at the ROLE Widget store for further reuse. Every bundle contains a name, a description and a list of apps. The user in *Graasp* can import OMDL files to create new spaces based on bundles. The file can be either uploaded manually or the user can browse through the bundles available in the Widget store with the *Graasp* bundle search interface and select the needed ones (Fig. 7). Once the bundle is selected, a new corresponding space is created. The space receives the name and the description of the bundle. The apps listed in the bundle are created and assigned to the space.
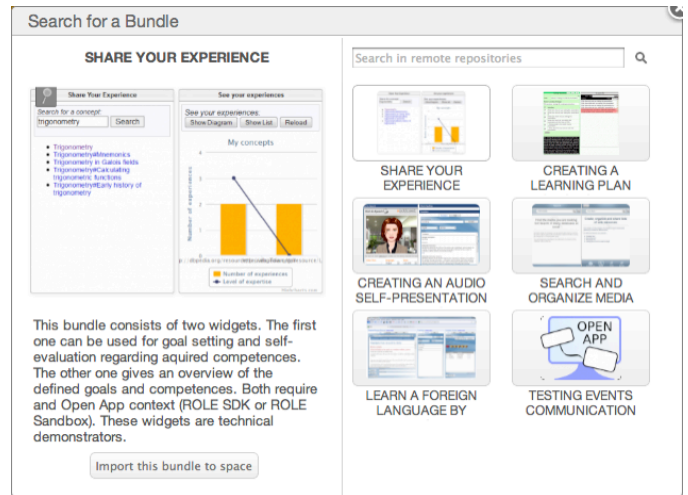

Fig. 7. The *Graasp* bundle search interface

A space created in *Graasp* can be shared with other people and with other platforms [6]. The space can be extracted from *Graasp* as a private or secret URL. This URL can be given to other users or open in any browser window. Alternatively, the space can be embedded into another Web environment as an iframe. The secret URL contains a secure token that authenticates users; so that users can accomplish the same actions they are permitted to do within a space in the *Graasp* interface. This approach allow users to access their spaces from another platform and share created spaces with people who do not necessarily have an account in *Graasp* (for example, a teacher sharing a learning activity space with students).

VII. RESOURCE SHARING WITH CREATIVE COMMONS

In order to facilitate the open sharing of resources and awareness, *Graasp* automatically proposes Creative Commons[8] licenses when content is made available in public spaces. This mechanism is extendable to OpenSocial apps source code when shared publicly to enable the repurposing of apps for personalization to other contexts. This feature is somehow also an enabler for a paradigm change in the exchange of information. By offering an easy way to add Creative Commons licenses, people are encouraged to share their resources freely in the cloud and to repurpose and valorize them in communities. Fig. 8 shows an example of a shared logo. The Creative Commons license is selected from the right-top license button and then appended to the resource automatically on the bottom-right.

In *Graasp*, spaces are either *public*, *closed* or *hidden*, and the embedded resources inherit these access rights. *Public* resources are accessible by anyone. *Closed* resources can be searched and recommended, but can only be viewed if access to the enclosing space is granted. *Hidden* resources are only visible to the invited members of the enclosing space. The selection menu for Creative Commons licenses is only available on resources belonging to a public space.

---

[7] http://omdl.org/

[8] http://creativecommons.org

Fig. 8. The Creative Commons license dialog attached to each public resource in *Graasp*

VIII. CONTEXTUAL SPACES AS AGILE KNOWLEDGE MANAGEMENT SOLUTIONS

The motivations for the development of online contextual spaces originated from the requirements elicited to support interaction in online learning communities [7] and aggregation of open educational resources and services for personal learning [8, 9]. As a consequence, the key features of the OpenSocial spaces are the

i. freedom for their users to choose the social media platform in which they are instantiated (providing the selected platform supports OpenSocial);

ii. ability to share through them resources across organization boundaries with members that can take various roles that can be changed over time;

iii. permeability to aggregate and repurpose both internal (legacy) and external (cloud) resources while enforcing intellectual property and access rights;

iv. plasticity that can be enabled by embedding alternative apps and skins to offer additional contextual services and interaction features.

All these features have attracted the interest of knowledge-oriented global institutions, organizations and enterprises, and especially non-governmental organizations (NGO) that are operating at a worldwide scale with distributed entities loosely integrated and relying on local hierarchical structures. NGO have also fast staff turnover and strong digital asset dependency that require effective knowledge management solutions. The ability to create contextual spaces managed in an easy and agile way by freelance experts in the field or in operation is hence very attractive to such organizations. Their main requirements are

i. being able to integrate aggregation and interaction spaces in matrix structures. A space corresponding to a given project like *Digging a Well in Ouagadougou* should be simultaneously located in a geographical hierarchy (Africa as example), in an activity one (like water management), and a department (like operation);

ii. managing access rights by roles, i.e. the owner of a space should be identified as *Project X Leader* rather than *John Smith* (or both) to enable an easy transfer of duties and resources to someone else when required;

iii. enabling fine identity management such as fusion of user profiles when people contribute to spaces with various credential (such as a general *gmail* account or social media platform OpenID) for consolidation purpose;

iv. facilitating changes in access rights, i.e. having no distinction between the intranet and the extranet or the internal and external cloud infrastructures (as example, when public documents are drafted internally and then made publically available, such feature ease the dissemination process);

v. guarantying the secrecy of some resources as NGOs are often dealing with content which can be politically sensitive. As a consequence, the physical and geographical location of the cloud resources should be fully defined and controlled.

vi. automating tagging of contributed content to ease search and recommendation without diverting contributors from their main tasks.

Such requirements are also very useful in the context of personal learning in which people often combine digital identities (as example when they move from one educational institution to another), where privacy issues are also critical (especially when dealing with young students), and where agile role management makes collaborative learning more effective. This section shows the interplay between personal learning and knowledge management in terms of supporting platforms and elicited requirements. In the future, design and validation between these two domains should be conducted more closely for cross-fertilization.

IX. CONCLUDING REMARKS

The usefulness of the space concept and the associated aggregation mechanism to populate them with resources gathered from the cloud have been demonstrated in this paper. Besides, the space personalization, providing alternative interaction modes with content, and the space sharing mechanism have been introduced. The permeability (ability to absorb content from the cloud) and the plasticity (ability to adapt the interaction mode to users' needs and add features through apps) of the spaces introduced in this paper pave the way for personal learning environments [6] fully constructed by the users for the users. Thanks to this agile personalization scheme, the *Graasp* platform introduced in this paper not only enables the instantiation of a PLE, but also enables the creation of dedicated spaces for project management, knowledge sharing or community support.

REFERENCES

[1] D. Gillet, S. El Helou, C.M. Yu, and C. Salzmann, "Turning Web 2.0 Social Software into Versatile Collaborative Learning Solutions," The First International Conference on Advances in Computer-Human Interaction (ACHI'08), Sainte Luce, Martinique, February 10-15, 2008.

[2] B. Charlier, F. Henri, D. Peraya, and D. Gillet, "From Personal Environment to Personal Learning Environment," In the Fifth European Conference on Technology Enhanced Learning (EC-TEL10), Workshop on Mash-Up Personal Learning Environments (MUPPLE), Barcelona, Spain, September 28th - October 1st, 2010.

[3] N. Dabbagh and A. Kitsantas, "Personal Learning Environments, social media, and self-regulated learning: A natural formula for connecting formal and informal learning," The Internet and Higher Education, 2012.

[4] E. Bogdanov, F. Limpens, N. Li, S. El Helou and C. Salzmann, "A Social Media Platform in Higher Education," Global Engineering Education Conference (EDUCON), Marrakesh, Morocco, April 17-20, 2012.

[5] E. Bogdanov, C. Salzmann, and D. Gillet, "Contextual Spaces with Functional Skins as OpenSocial Extension," In 4th International Conference on Advances in Computer-Human Interactions, 2011.

[6] S. Sire, E. Bogdanov, M. Palmér, and D. Gillet, "Towards Collaborative Portable Web Spaces," In 4th European Conference on Technology Enhanced Learning (EC-TEL) - Workshop on Mash-Up Personal Learning Environments (MUPPLE), 2009.

[7] Y. Rekik, D. Gillet, S. El Helou, and C. Salzmann, "The eLogBook Framework: Sustaining Interaction, Collaboration, and Learning in Laboratory-Oriented CoPs," International Journal of Web-Based Learning and Teaching Technologies, Vol. 2, No. 3, pp. 61-76, 2007.

[8] D. Gillet, "Tackling Engineering Education Research Challenges: Web 2.0 Social Software for Personal Learning," International Journal of Engineering Education, Special Issue on Applications of Engineering Education Research, Vol. 26, No. 5, pp. 1134-1143, 2010.

[9] D. Gillet, E. Law, and A. Chatterjee, "Personal Learning Environments in a Global Higher Engineering Education Web 2.0 Realm," 1st IEEE Engineering Education Conference (EDUCON), Madrid, Spain, April 14-16, 2010.

Fig. 9. Shared contextual space created in *Graasp* and integrating resources gathered from the cloud in one click using *GraaspIt!*, such as YouTube videos, SlideShare presentations, ROLE Widgets, Web pages or pdf documents with previews