Go-Lab

Global Online Science Labs for Inquiry Learning at School

Collaborative Project in European Union's Seventh Framework Programme

Grant Agreement no. 317601

Deliverable D5.7

Release of a sustainable version of the Go-Lab portal and the associated end-user solutions

Editors: María Jesús Rodríguez-Triana and Adrian Holzer (EPFL)

Date: 31st October 2016

Dissemination Level: Public

Status Final

(c) 2016, Go-Lab consortium

The Go-Lab Consortium

Beneficiary Number	Beneficiary name	Beneficiary short name	Country
1	University Twente	UT	The Netherlands
2	Ellinogermaniki Agogi Scholi Panagea Savva AE	EA	Greece
3	École Polytechnique Fédérale de Lausanne	EPFL	Switzerland
4	EUN Partnership AISBL	EUN	Belgium
5	IMC AG	IMC	Germany
6	Reseau Menon E.E.I.G.	MENON	Belgium
7	Universidad Nacional de Educación a Distancia	UNED	Spain
8	University of Leicester	ULEIC	United Kingdom
9	University of Cyprus	UCY	Cyprus
10	Universität Duisburg-Essen	UDE	Germany
11	Centre for Research and Technology Hellas	CERTH	Greece
12	Universidad de la Iglesia de Deusto	UDEUSTO	Spain
13	Fachhochschule Kärnten – Gemeinnützige Privatstiftung	CUAS	Austria
14	Tartu Ulikool	UTE	Estonia
15	European Organization for Nuclear Research	CERN	Switzerland
16	European Space Agency	ESA	France
17	University of South Wales	USW	United Kingdom
18	Institute of Accelerating Systems and Applications	IASA	Greece
19	Núcleo Interactivo de Astronomia	NUCLIO	Portugal
20	Cardiff University	CU	United Kingdom

Contributors

Name	Institution
Yiwei Cao, Nils Faltin	IMC
Adrian Holzer, Denis Gillet, María Jesús Rodríguez-Triana, Andrii Vozniuk, Alex Wild, Wissam Halimi	EPFL
Alexandros Trichos, Panagiotis Zervas	CERTH
Pablo Orduña	UDEUSTO
Lars Bollen, Ton de Jong, Jacob Sikken, Anjo Anjewierden, Ellen Wassink-Kamp	UT

Internal Reviewers

Name	Institution
Sven Manske, Ulrich Hoppe	UDE
Rosa Doran	NUCLIO

Legal Notices

The information in this document is subject to change without notice.

The Members of the Go-Lab Consortium make no warranty of any kind with regard to this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Members of the Go-Lab Consortium shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

The information and views set out in this deliverable are those of the author(s) and do not necessarily reflect the official opinion of the European Union. Neither the European Union institutions and bodies nor any person acting on their behalf may be held responsible for the use which may be made of the information contained therein.

Executive Summary

This deliverable presents the releases of the sustainable version of the Go-Lab Portal and the associated end user solutions following the releases presented in D5.6 on M36 and in D5.5 on M32. In this deliverable we elaborate on the features that were added or modified since the releases of D5.6, especially with the focus on the sustainability of the Go-Lab Portal, the ILS Platform (also sometimes referred to as the ILS Authoring Platform), the App Composer, and also the other end user solutions developed in the project, such as the inquiry learning apps and the tutoring platform. In a nutshell, this deliverable summarizes the main changes that have been integrated in the different Go-Lab platforms in Year 4.

Go-Lab Repository. Since the final releases were launched at the end of M36, many online labs, apps, and ILSs have been added to the Portal. Furthermore, we focused on providing further support for *teachers*, e.g., user manuals, community forum, as well as *lab owners* and *app creators* through, for example, enhanced user-friendly input forms and an improved search feature. More user roles are implemented in the repository, based on a conceptual user scheme (see Figure 2-9). This helps content creation and quality control. More details about these developments can be found in Section 2.

ILS Platform. The main changes in the ILS platform have to do with user empowerment. That is, there is a focus on providing more support for users to autonomously work on the platform. For instance, such support consists of more help and internationalisation, improved warning messages, improved search and recommendations, and improved awareness through activity visualisation. Furthermore, as the number of users grows, changes involved scalability improvements to sustain a larger user base. More details about these developments can be found in Section 3.

App Composer. The main changes of the App Composer aim at supporting translators in finding applications to be translated and at the integration with the Go-Lab portal through automatically changing the existing metadata related to what apps are available in which languages. The App Composer also provides statistics of its usage including what language translations are missing so that the Go-Lab administrators can request further translations to people who already provided translations for a language. More details about these developments can be found in Section 4.

Apps. Concerning the apps and supportive scaffolds, the recent developments addressed stability, performance, minor bug fixes, and usability improvements. Changes made to other components (e.g.', ILS Platform, App Composer) have been implemented in the apps as well, when necessary. The Report tool, which was in a prototype state previously, has been finalized and released to the Go-Lab Portal. A number of new learning analytics tools help to interpret and understand the students' results and artefacts. More details about these developments can be found in Section 5.

Table of Contents

Executive Summary 4
Table of Contents 5
1. Introduction 7
1.1 Structure of this deliverable 7
1.2 Numbers and facts 7
2. The Go-Lab Repository 9
2.1 Functional requirements 9
2.1.1 Creating, publishing, and using ILSs 9
2.1.2 Search labs, apps & ILS 11
2.1.3 Getting support 12
2.1.4 Publishing labs 13
2.1.5 Managing tutorials 14
2.1.6 User management 15
2.1.7 Tracking user activities 15
2.2 Non-functional requirements 17
2.2.1 Interoperability 17
2.3 Summary 18
3. The ILS Platform 19
3.1 Functional requirements 19
3.1.1 Creating ILSs 19
3.1.2 Publishing and Copying ILSs 21
3.1.3 Using ILSs 22
3.1.4 Searching users and content 23
3.1.5 Activity tracking and User privacy 24
3.1.6 Recommendations 25
3.1.7 User management 26
3.2 Non-functional requirements 29
3.2.1 Usability 29
3.2.2 Scalability 31
3.2.3 Internationalisation 32
3.3 Summary 32
4. The App Composer 34
4.1 Functional requirements 34
4.1.1 Publishing non OpenSocial apps in the repository 34
4.1.2 Translation of apps 34
4.1.3 Translators finding apps to be translated 34
4.1.4 Automatic update of the language metadata in the Go-Lab Portal 35
4.2 Non-functional requirements 35
4.2.1 Scalability 35
4.2.2 Translation Maintenance 35

4.3 Summary 36
5. Inquiry Learning and Learning Analytics Apps 38
5.1 Latest App Developments 38
5.1.1 Reporting Tool 38
5.1.2 Survey App 39
5.1.3 Concept Map Dashboard 41
5.1.4 Group Formation App 42
5.2 Code repositories 44
5.3 Summary 44
6. Conclusion 45

1. Introduction

In this deliverable, we will describe the new features and new content to support the sustainability of the Go-Lab Portal and the App Composer. Since the Go-Lab project is in its last year, our work focused on supporting user communities from a technical point of view, simplifying the use of the platforms for different user groups, and eliminating maintenance tasks. Overall, this aims at updating the Go-Lab Portal and the App Composer in such a way as to provide sustainable, easily-maintained platforms for long term use within the Go-Lab user communities.

1.1 Structure of this deliverable

This deliverable is based on the same structure as the previous deliverables in Work Package 5 (D5.2, D5.4, and D5.6). It is extended with the other end-user solutions designed and developed in the project, namely inquiry learning apps and the Go-Lab Tutoring Platform.

1.2 Numbers and facts

Below is the status of the platforms as of September 30, 2016.

The Go-Lab Portal

- The Go-Lab Repository (golabz.eu)
  - Labs: 401
  - Apps: 39
  - ILSs: 396
  - Videos: 23
  - Usage: around 10’000 sessions per month
- The ILS Platform (graasp.eu)
  - Registered Users (teachers): 9770
  - ILSs created: 13812
  - Average number of users per day (from October 2015): 142, reaching peaks of more than 500 users during the course.

The App Composer

- Embedder:
  - Labs: 263 labs
- Translator:
  - 58 labs, apps and tools use the standard translation mechanism -OpenSocial i18n- used by the App Composer.
    - An average of 71 messages per application due to few large apps. The median is 25 messages per application.
  - Users: 41 translators of which everyone translated more than 100 messages.
  - Languages: 21,545 messages translated by humans in 30 languages. In 16 of them there are multiple applications 100% translated.
  - Public stats:
    - http://composer.golabz.eu/translator/stats/status
    - http://composer.golabz.eu/translator/stats/golabz

The Smart Gateway (see D4.7)

- Labs: 94 labs
- Translations:
    - 27 of the labs can be translated using the App Composer
    - 27 of the labs cannot be translated using the App Composer but translations are provided
    - 40 of the labs come from providers (such as concord[1]) that do not support translations neither they provide internationalisation

The Tutoring Platform

- Forums: 6 with about 160 posts
- Users: ca. 370 registered users

[1] http://www.concord.org

2. The Go-Lab Repository

Based on the previous functional and non-functional requirements and in line with deliverable D5.6, this chapter lists the new features. In addition, the Go-Lab Tutoring Platform as an independent Go-Lab resource has been finalized in WP4 by M36. Since it also contributes to community building and to the sustainability of the Go-Lab Repository, we also report the progress of the Tutoring Platform in this chapter (see D6.7 for more details about the Tutoring Platform).

2.1 Functional requirements

2.1.1 Creating, publishing, and using ILSs

Integrate scenarios to create a scenario-based ILS from the Go-Lab Repository

Although scenario-based ILS creation has already been available in Graasp in the previous release, this release provides extra details to inform teachers about this new feature and guide them in its usage (see Figures 2.1 and 2.2).

Users (mainly teachers) are supported to create ILSs in several different ways including the new features of creating an ILS based on scenarios from the repository.

- From the repository (Golabz)
  - On an online lab page by clicking “Create an Inquiry Space”
  - On an ILS page by clicking “Copy & use this Inquiry Space”
  - On a scenario page by clicking “Use this scenario”
- From the ILS Platform (Graasp)

Figure 2.1. The scenario description page in Golabz.

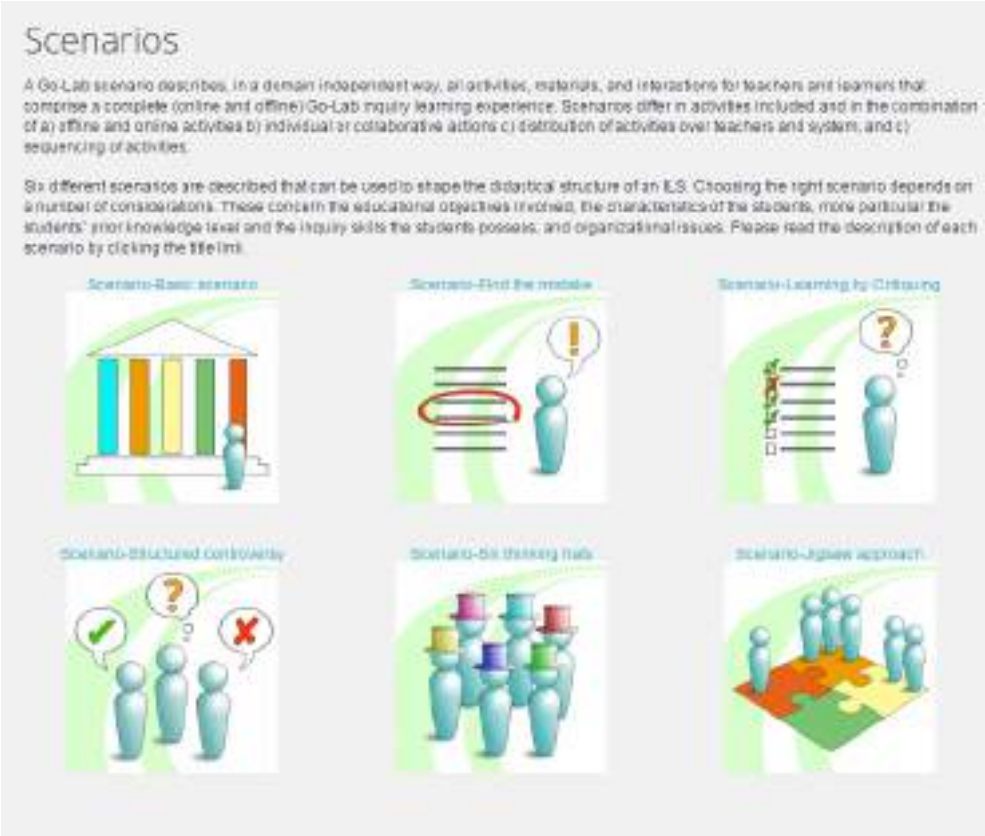

Figure 2.2. The catalogue page of all scenarios.

Show Java/Flash labs

According to the feedback from many teachers in the framework of WP3 workshops and activities, some teachers had encountered problems with some Java or Adobe Flash based labs, which led them to avoiding those labs. So for each lab, an additional field about the underlying technology has been added. Figure 2.3 shows a Java technology enabled lab showing teachers the corresponding technological information. Note that as a policy, we do not add new Java labs anymore.

Figure 2.3. A Java technology enabled lab.

Listing all spaces which use certain labs or apps

Again based on teachers' feedback, we have extended the feature of displaying for each app and lab the list of inquiry spaces which make use of them. Besides the random list of three ILSs, an additional link "show all spaces" has been placed. And users can see how many spaces in Golabz use the lab. An example of the Splash lab is illustrated in Fig. 2.4. With this feature, the lab owners and app creators can get a good overview about how popular their apps or labs are, in terms of how many ILS use this resource. As of September 30, 2016, 186 published spaces have been using the app Hypothesis Scratchpad (http://www.golabz.eu/related-ils/25) and 99 spaces have been using the app Concept Mapper (http://www.golabz.eu/related-ils/7).

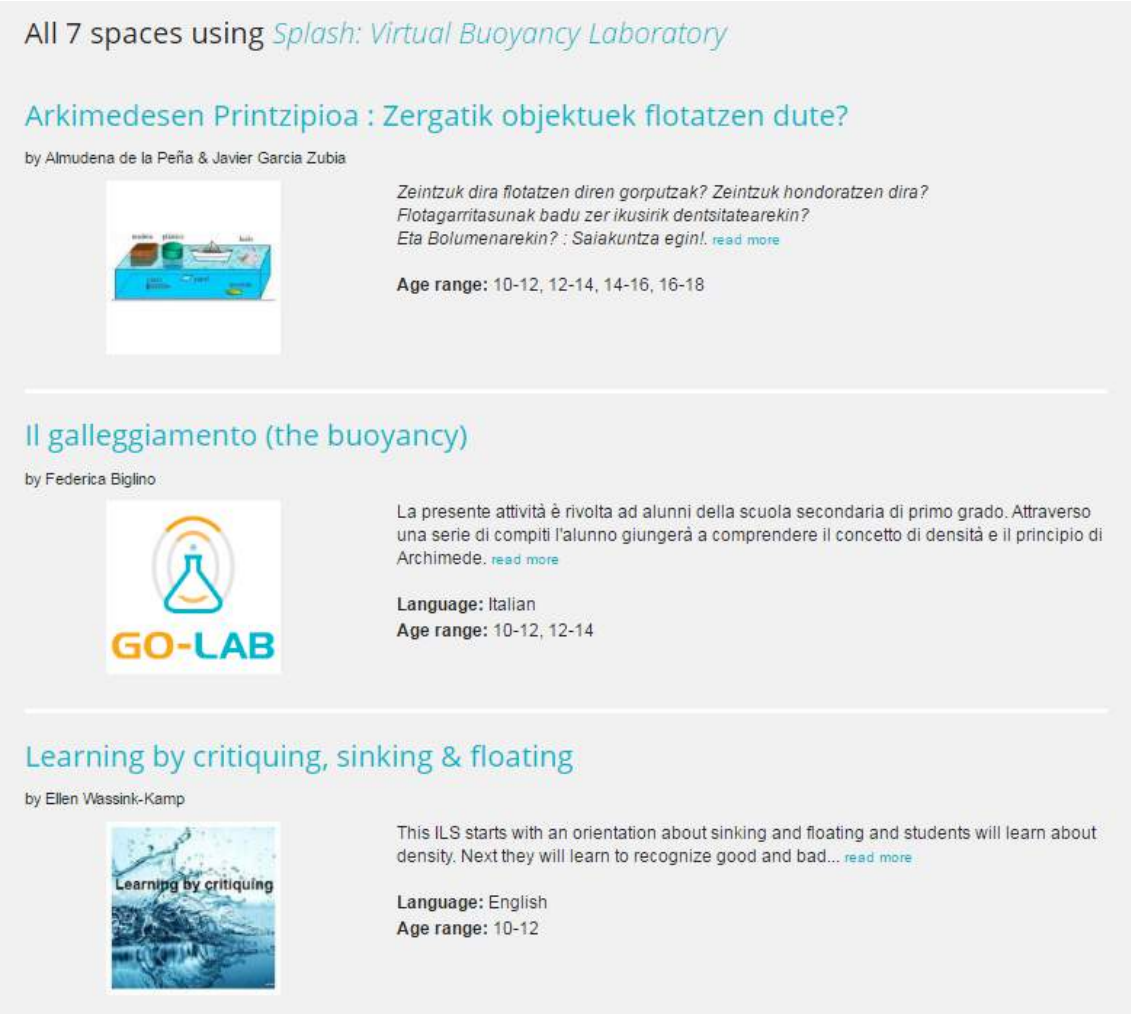

Figure 2.4. A page to list all ILS which use the lab Splash.

2.1.2 Search labs, apps & ILS

Display Big Ideas in a better way

According to the feedback from WP2's interactions with the teacher communities, few teachers use the Big Ideas to search for labs or ILS. Thus, additional icons have been added in the sidebar to visually support a faceted search related to Big Ideas as shown in Figure 2.5.

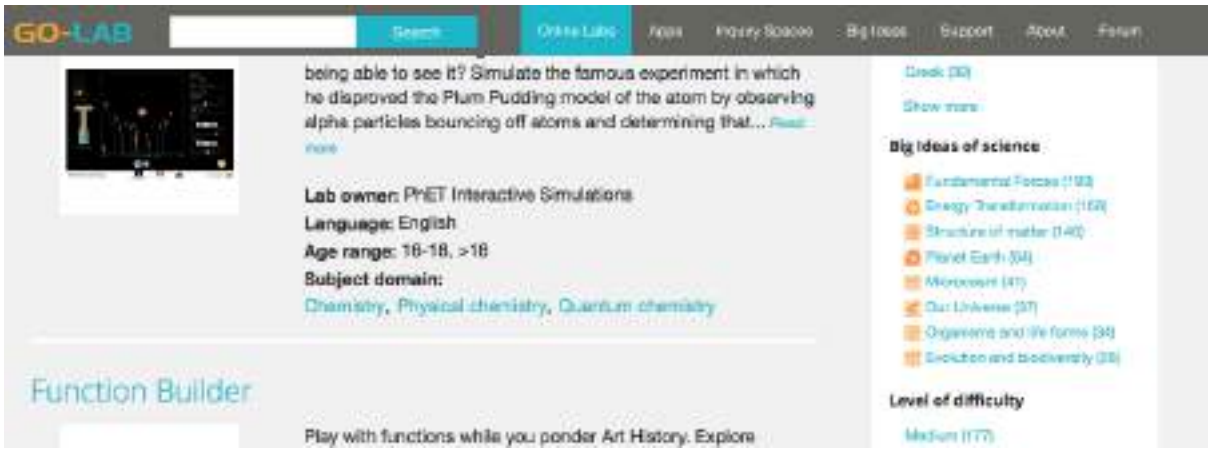

Figure 2.5. Big Ideas are displayed in the faceted search.

Advanced search

With the growth of the learning resources (labs, apps, and ILS) in Golabz, users may get a long list of search results. Thus, the advanced search has been implemented so that users are able to

- search resources according to its content type, labs, apps, ILS, videos, tutorials, and user contributors, such as lab owners and ILS creators;
- and exclude unnecessary content types like lab booking info.

Figure 2.6 shows the advanced search user interface. Users can choose the types by clicking the check boxes.

Figure 2.6. Advanced search of Golabz.

2.1.3 Getting support

Support of the Community Forum

To provide a smoother user experience, we provided links to the community forum (a part of the Tutoring Platform) directly from Golabz through the Forum tab. Hence teachers who visit Golabz can now more easily join the forum to meet with the Go-Lab teacher community and discuss issues and experiences. Figure 2.7 shows the forum page with different forum topics.

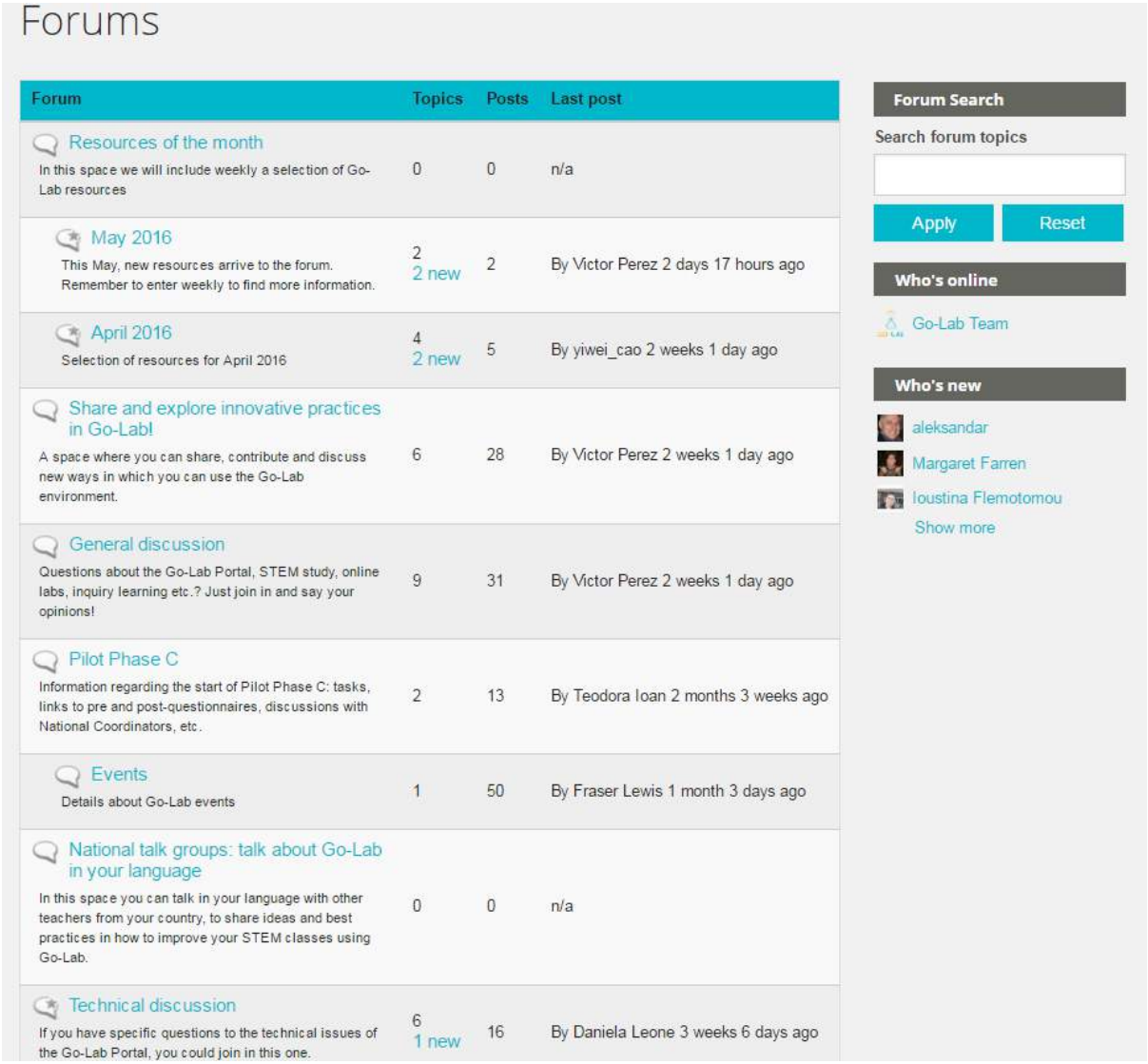

Figure 2.7. Go-Lab community forum hosted in the Go-Lab Tutoring Platform.

User manuals

User manuals have been added on the support page http://www.golabz.eu/support. This includes two parts: Go-Lab Classroom Scenarios Handbook and Go-Lab Teachers' Support Manual. The content had also been translated to many other European languages besides English, including Basque, Bulgarian, Catalan, Dutch, Estonian, German, Greek, Portuguese, Romanian, Spanish, and Turkish.

2.1.4 Publishing labs

Improvement of the workflow for publishing a new lab

We have encountered problems and received feedback from lab owners about lacking metadata and missing Open Social gadget XML links (a gadget is a Go-Lab app or Lab interface and is sometimes referred to as widget as well). The latter is so critical that the published lab without the gadget link may possibly not be used in an ILS. However, this could not be solved directly by making this field mandatory, because this may lead to a loss of motivation to publish on the part of lab owners if they cannot save the input content easily. To deal with this dilemma, we have added more information about how to create the widget (as

shown in Figure 2.8). The field of the gadget link is also marked in red with detailed explanations. The lab input form has also been improved by adding more explanation to each field and change the “educational objectives” metadata field into multiple choice input instead of free-text input.

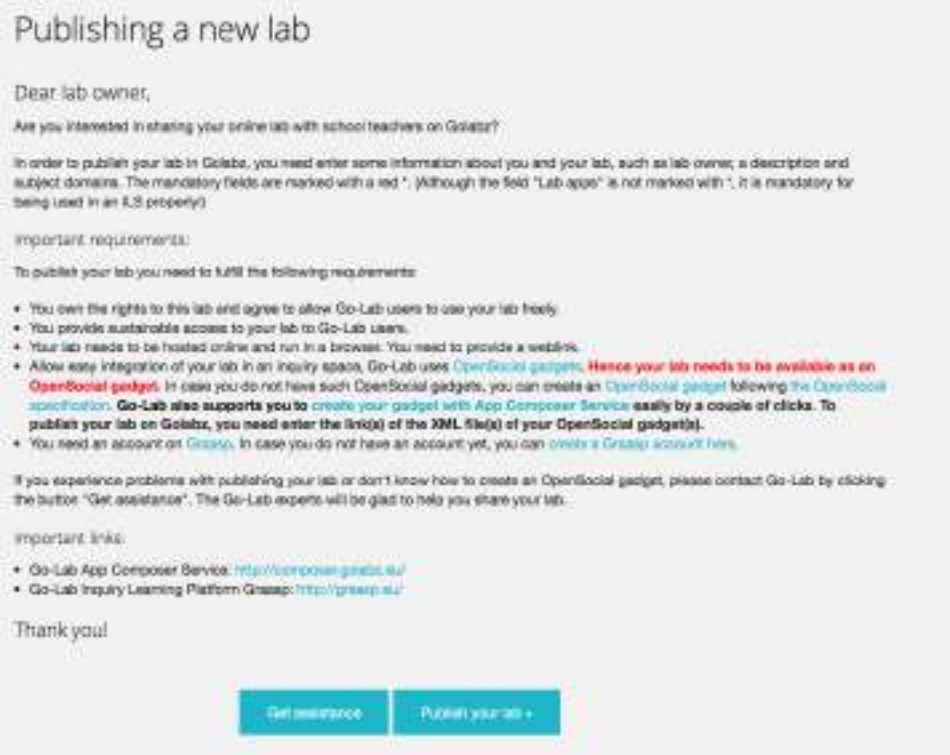

Figure 2.8. Information for lab owners about the lab publishing process.

Update lab/app language automatically with the App Composer

Since more and more lab owners use the App Composer to create, manage and translate their labs, we synchronised the language information for labs between the Go-Lab Repository and the App Composer. This is how it works: The App Composer calls the RESTful service from Golabz to call the lab languages, and compares them with the language info of the same lab in the App Composer. If there is a difference, it will be put into a change list in JSON format. Golabz calls the change list every hour as a cron task and updates the language information accordingly. For those labs that are not hosted in the App Composer, the lab owners still need to add the lab language information manually.

In addition, the language information of apps are also displayed in Golabz which is retrieved in real-time from the App Composer. More information about this new feature will also be described in the latter chapter “The App Composer”.

2.1.5 Managing tutorials

Tutorial videos

In addition to online labs, learning apps and inquiry spaces, the hosting of tutorials is also getting more important. Many teachers informed the Go-Lab team that they had questions about how to use a certain functionality of the platform. Thus, Golabz has specified a role called “Video manager” to be responsible for the management of tutorial videos with easy user interface. This role can also be given to multiple users, so that a group of super users or Go-Lab owners could collaborate on it.

2.1.6 User management

As mentioned in the use case diagram in D5.6, users with different roles interact with the platform. A conceptual user role scheme is specified for the Go-Lab Repository. Figure 2.9 shows the lab owner role scheme. App creator and teacher (ILS publisher) share the same concept. This feature has been implemented for lab owners and app creators in the repository.

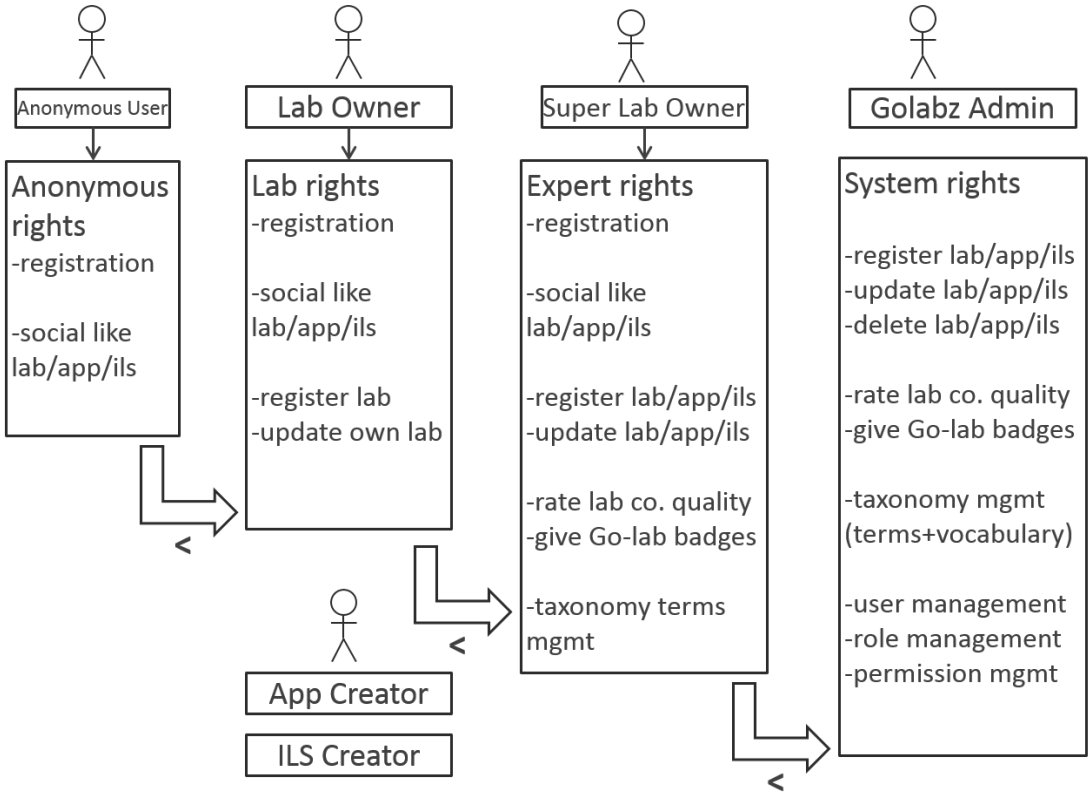

Figure 2.9. The Go-Lab user role scheme.

In addition, Golabz has also specified the user roles of “video manager” who is in charge of creating, updating and managing all video tutorials for users. The Go-Lab Tutoring Platform specifies the role of “forum moderator” who creates, moderates, and manages the forum and its topics. Usual authenticated user could only add new forum topics and posts, without the access right to create a new forum.

2.1.7 Tracking user activities

Notification emails for super lab owners

The Go-Lab Repository supports lab owners in publishing their labs easily, which enriches the content more rapidly. This can come at a cost since some labs may be of low quality because the lab owner may not be familiar with all metadata of the repository. In order to improve the quality of lab content, a feature has been developed to notify the super lab owners as soon as a new lab is added. The super lab owners can check the content, update and improve the lab, and contact the lab owner for more information if needed.

Integration of Piwik statistics

Lab owners, app creators, and ILS creators will see the statistics of their resources in different dashboard views to raise their awareness about users visiting the page (see Figure 2.10).

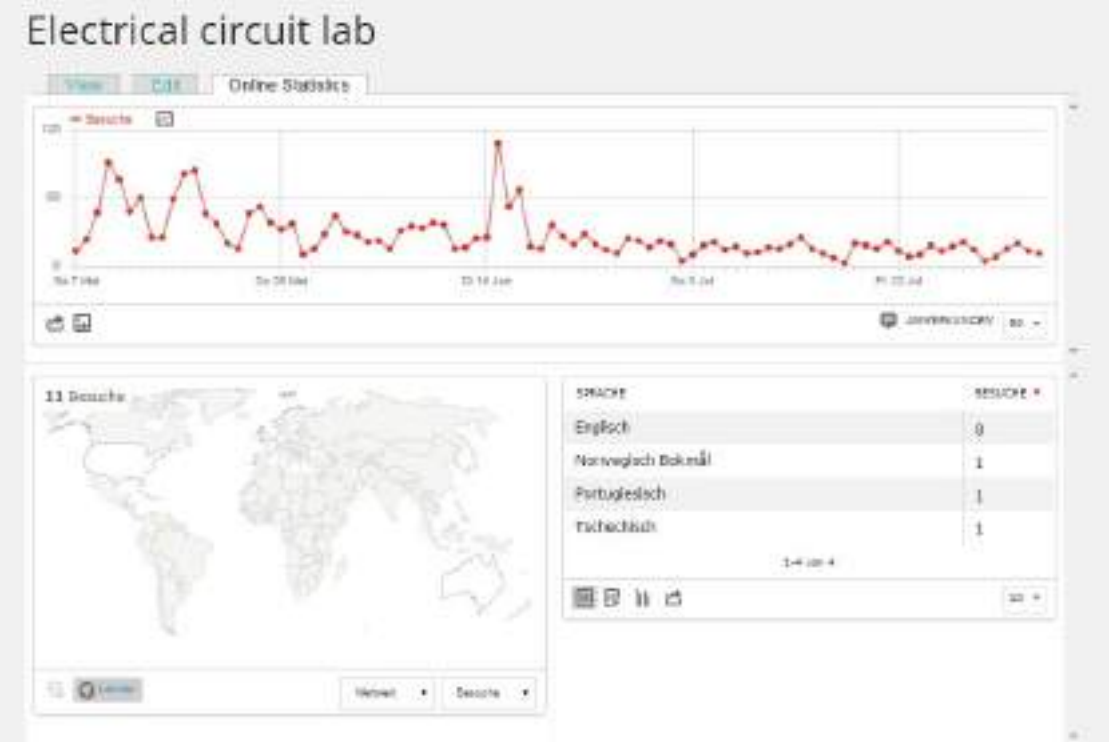

Figure 2.10. The statistic dashboard for lab owners.

Integration of usage statistics

Lab owners can also view how their lab was used in ILSs. Such awareness can further motivate them to maintain and update their content (see Figure 2.11).

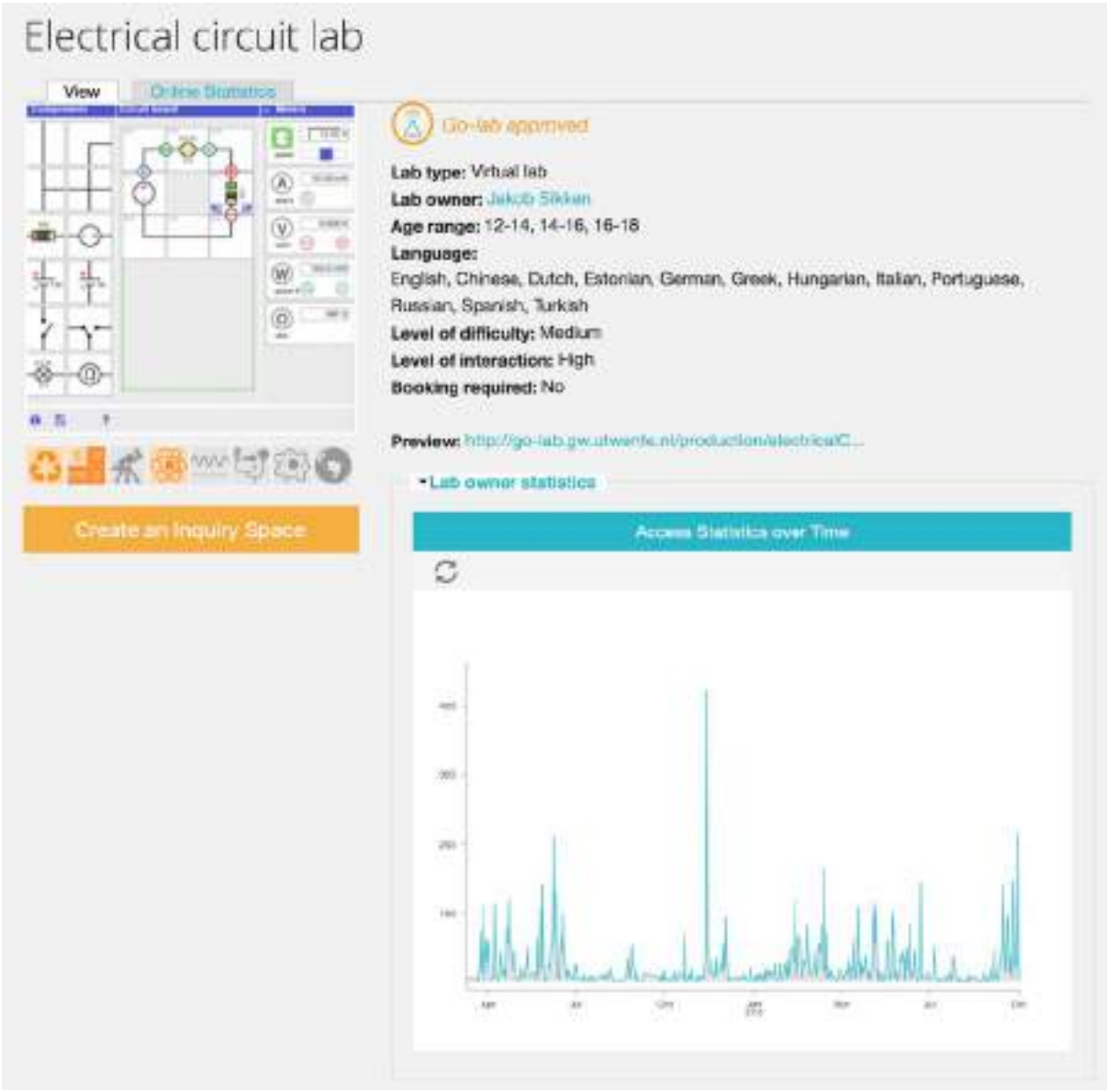

Figure 2.11 The usage statistic for lab owners.

2.2 Non-functional requirements

2.2.1 Interoperability

Extended RESTful service for external platforms

The Go-Lab Portal aims at attracting and providing better use for a wider user community. Cooperation and collaboration with other lab providers and platforms are crucial for sustainability. In most cases, the Go-Lab team gets cooperation requests from other platform holders. The Go-Lab Repository has extended the existing RESTful services to offer metadata to different lab owners and learning portals. Such metadata can be retrieved through the following links:

- Published apps: http://www.golabz.eu/rest/apps/retrieve.json
- Published labs: http://www.golabz.eu/rest/labs/retrieve.json

- Published ILSs: http://www.golabz.eu/rest/ils/retrieve.json

As a matter of fact, the following learning portals use our RESTful services:

- ViSH is a social and collaborative site for creating and sharing knowledge resources for the realization of education activities. They have request on Go-Lab Repository RESTful service to list all ILS at real time. (Link: http://vishub.org/categories/214)
- Kennisnet is a public organization for education and ICT ranging from primary school education to professional training based in the Netherlands. They will integrate Go-Lab learning resources into Edurep, a digital learning material repository. (Link: https://www.kennisnet.nl/edurep/)
- AMIGO (smArt steM learnInG ecOsystem) project has requested support from Go-Lab in order to integrate Go-Lab tools available in the repository into AMIGO learning environment for STEM education. The cooperation is in progress; technical implementation is foreseen for October-November 2016.

The Go-Lab Portal has also cooperated with many lab owners e.g. PhET who has been delivering the repository new labs with advanced technologies.

2.3 Summary

A sustainable platform must attract content providers and users. In a nutshell, the aforementioned features contribute to this goal through the following mechanisms:

- Through **enhanced lab publishing** mechanisms, support and analytics, we make it easier for lab owners to publish labs and to monitor their usage.
- Through **increased teacher support,** through manuals, forums, and videos, we make it easier for them to use the system and to publish their own ILSs.
- Via the aforementioned notification mechanism, and the super-user management roles, the Go-Lab team is kept informed when new resources are added. Such features are also helpful for **quality control**.

3. The ILS Platform

The ILS Platform (also sometimes referred to as the ILS Authoring Platform) is accessible at http://graasp.eu. We also call it *Graasp,* in short. Following the changes specified in G5.5 towards a sustainable version of Graasp, this section describes all the new features and changes included which were not already present in the last deliverable (i.e., D5.6).

3.1 Functional requirements

3.1.1 Creating ILSs

In order to enrich ILS creation, the sustainable version of Graasp allows teachers to:

- Add **hints** for the students that can be displayed on student request. This feature allows teachers to add items to the ILS and set them as hints (see Figure 3.1), which are hidden them in the standalone view behind an appropriate button (see Figure 3.2). When students press the button, the item appears. Such hints allow teachers to include extra material in their ILS without making the ILS cluttered.

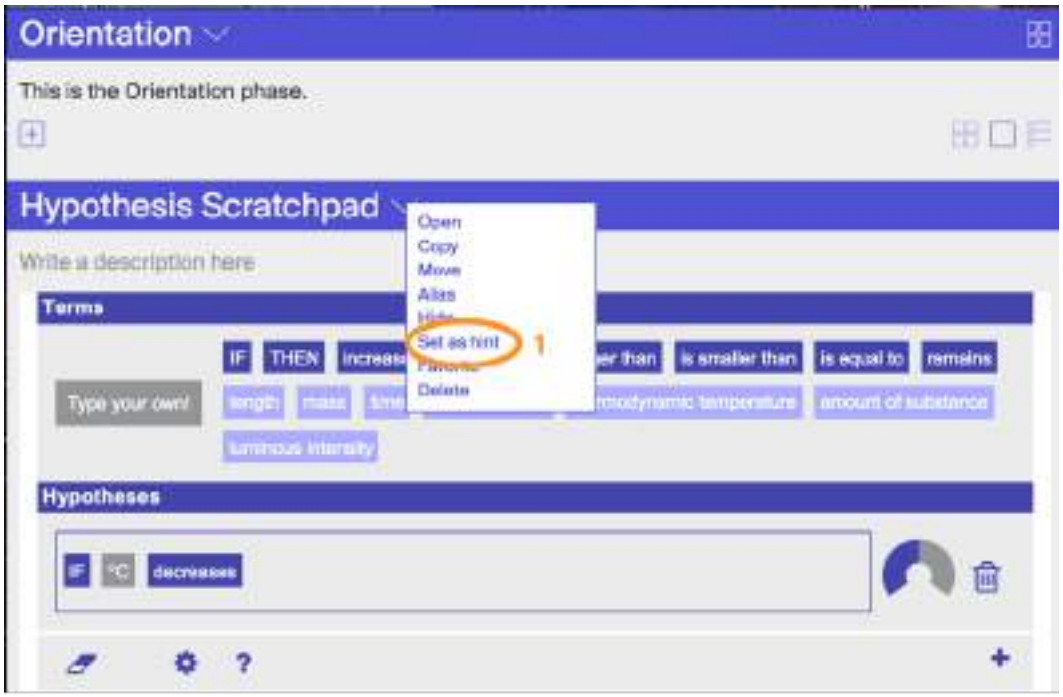

Figure 3.1. Adding hints in the authoring view.

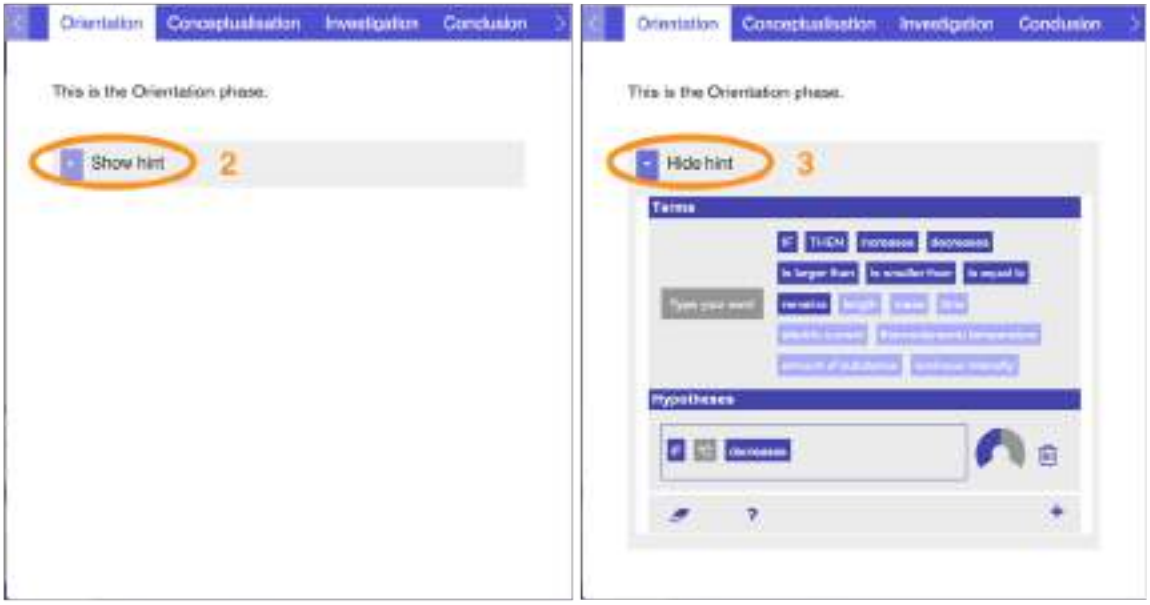

Figure 3.2. Visualization of a close an open hint in the standalone view.

- Add **discussion threads** in the ILS enabling the discussion about different topics with other members (owners and editors). As Figure 3.3 illustrates, in order to add a new discussion thread, the user just need to open the "Create item" button and choose the option "Create discussion". A topic can be seen as a forum discussion between members as it is shown in Figure 3.4. This discussion can be moved to another space and possibly be made public if desired (it could be added to a FAQ space for instance if it is relevant). The members of the space can also like/dislike the messages and order them according to the popularity or the date when they were posted.

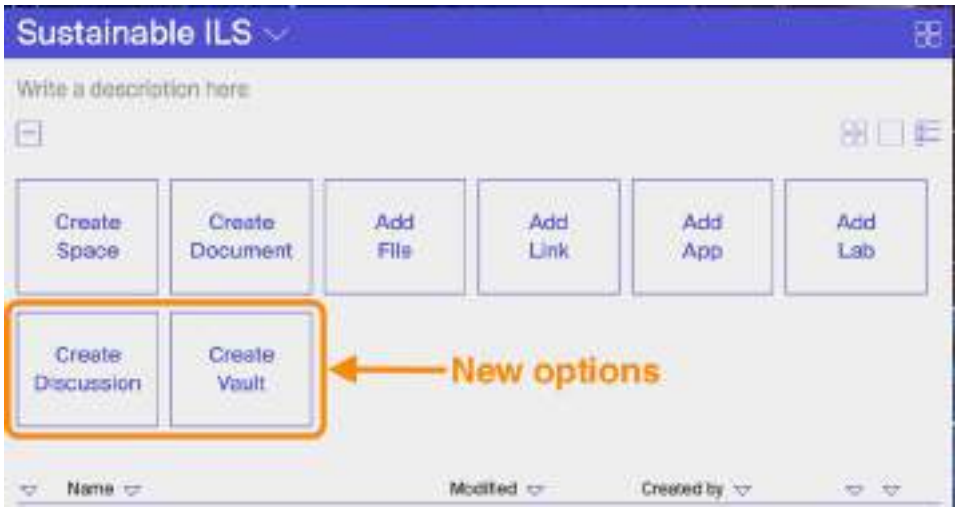

Figure 3.3. Extended "Create item" menu including discussions and vault spaces.

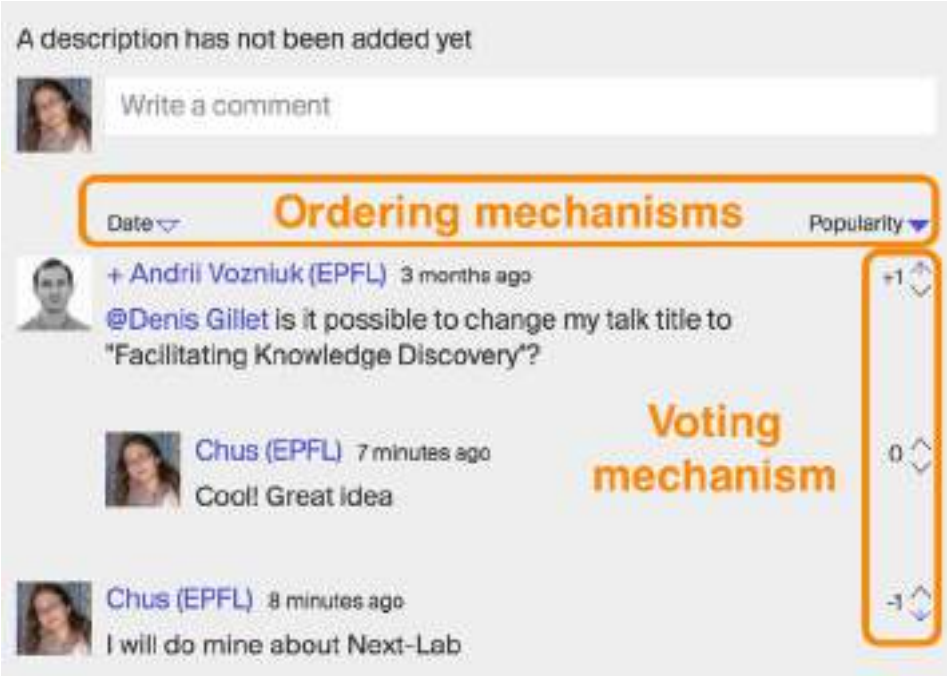

Figure 3.4. Visualization of a discussion thread.

- Re-create a **Vault** space after such a space has been deleted or moved. Since space owners can manage all items in a space, they can decide to delete or move the vault space (e.g., in order to clean it for a new course or to keep a backup of the students' outcomes). Since the Vault space is used as data repository for apps, the lack of a Vault space may result in some apps not working anymore. To warn the teacher of this issue a message is displayed when the user attempts to delete the Vault (see Section 3.2.1 for further details about such messages). Furthermore, the teacher will be able to recreate one using the create vault button as depicted in Figure 3.3.

3.1.2 Publishing and Copying ILSs

To encourage teachers to publish and update their ILSs, the sustainable version of Graasp allows teachers to:

- **Republish** their ILSs or create a new published version. In case of republishing, the metadata provided by the user in Go-Lab Repository the first time the ILS was published is reused in order to minimize the republishing effort. Nevertheless, the teachers are able to modify the metadata whenever necessary in the repository.

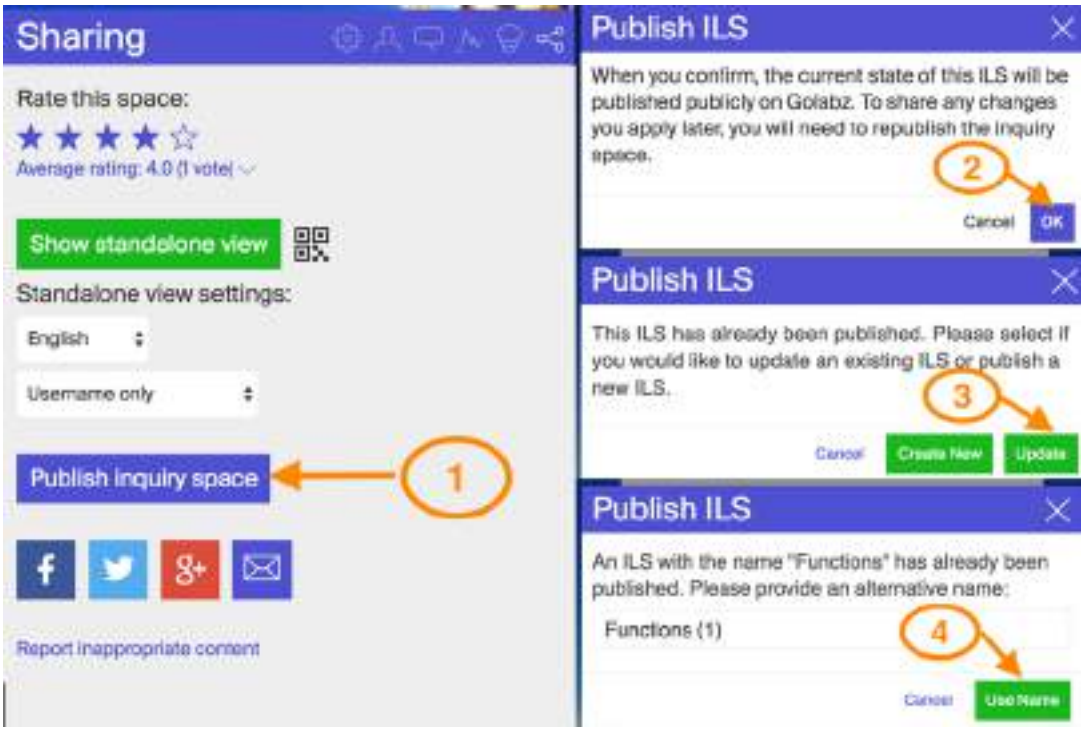

Figure 3.5. Republishing flow: the 4 steps required to republish an ILS.

3.1.3 Using ILSs

In order to support ILS usage, the sustainable version of Graasp offers:

- A **QR code** to share the standalone view without the need to use the URL. With this option a teacher can display the QR code on the projector and students can use a standard QR scan app on their iPads to access the ILS. This functionality is available in the sharing tab as it is shown in Figure 3.6.

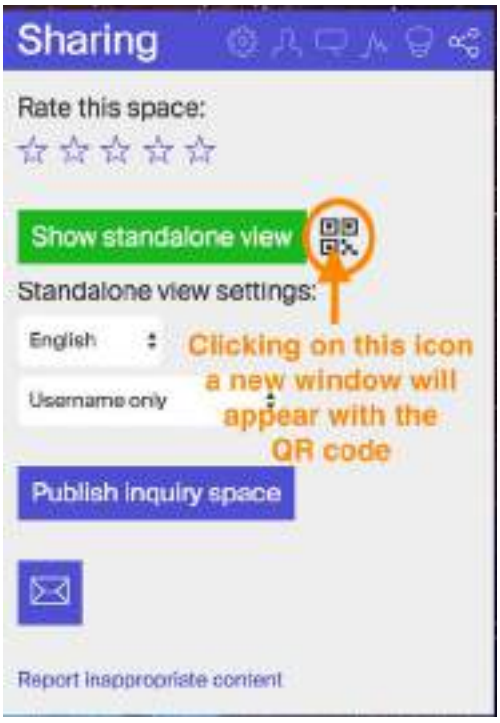

Figure 3.6. Sharing an ILS using a QR code.

- An **export option** to save the artefacts generated by the students in an ILS. Opening the student work in review mode, a printer icon appears next to the ILS title (see Figure 3.7). Clicking on the printer, a menu pops up offering the options of: a) exporting the material in PNG or PDF; and b) print the whole ILS content or simply the learning outcomes generated by the apps. Note that such an option has limitations since it will not be able to print the content of apps or labs that do not support the appropriate print instructions. This is typically the case for third party solutions.

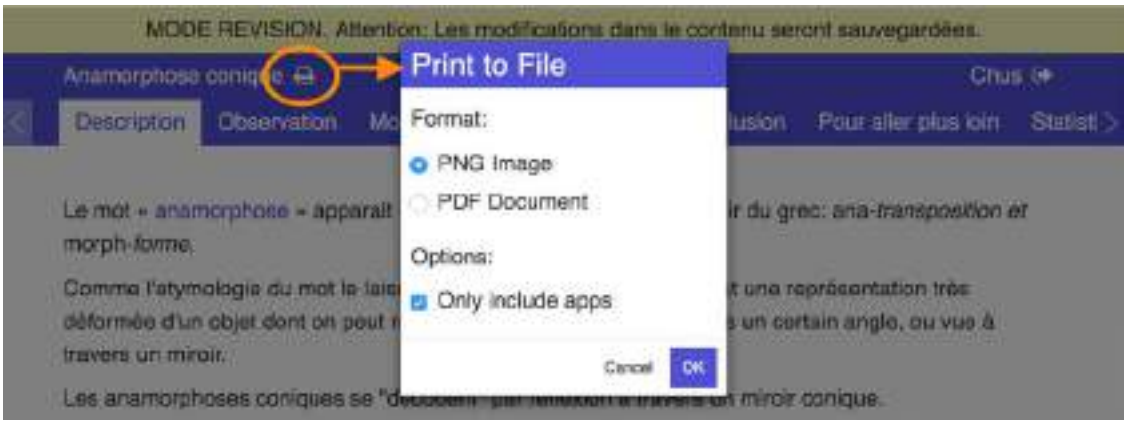

Figure 3.7. Printing and ILS and its content generated by a student as a file.

3.1.4 Searching users and content

In order to support searching of Labs and ILS, the sustainable version of Graasp includes:

- Handling of **misspelled words**, underlying those words that are not considered as correct, as it is illustrated by Figure 3-8.

Figure 3.8. Identification of misspelled words in the search.

- Provision of **faceted search** functionalities for users and content (alias, applications, resources, spaces or topics) as well as location (either in the whole platform or in the current space). Furthermore, it displays the number of elements found in each category, the location of the items and, if it is available, a brief description. In order to interact with the items, clicking on the arrow next to the title, a menu appears offering the following options: open, copy, alias and favourite.

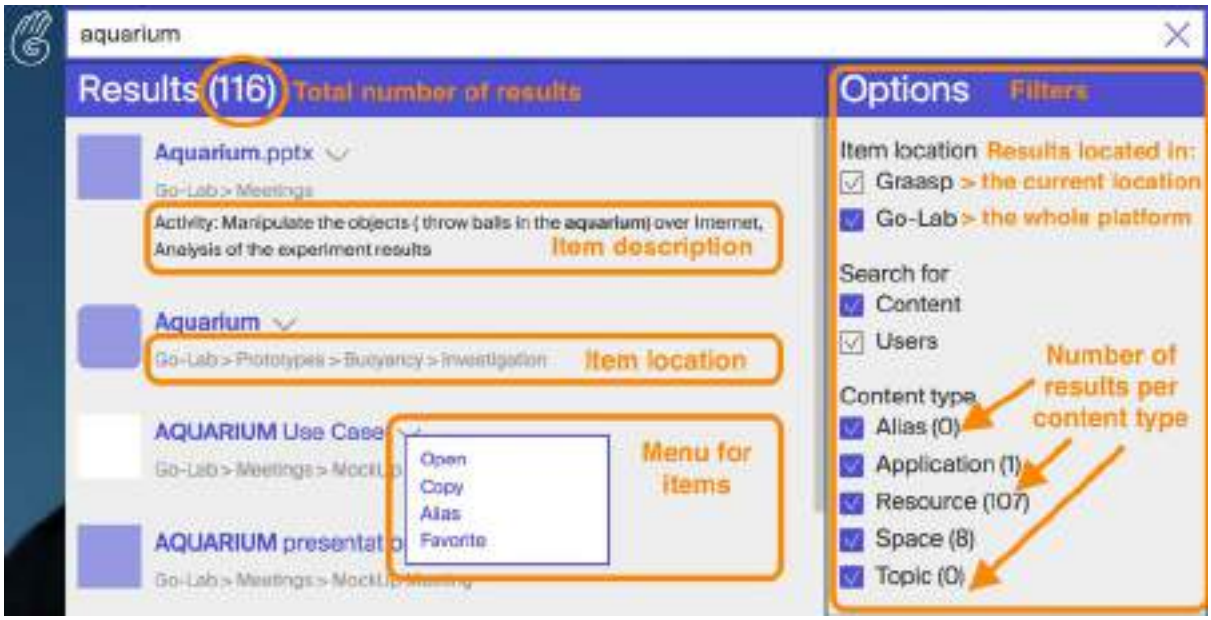

Figure 3.9. New functionalities added to the search.

3.1.5 Activity tracking and User privacy

To support activity tracking and user privacy, the sustainable version of Graasp provides:

- **LA dashboard** based on Kibana and Elasticsearch[2]. This interactive dashboard allows teachers to visualize all activities that occurred in a certain space and dig into the data without the need to add an app to the space. To open this dashboard, the use has to click on the pie chart icon on the title bar of the space as shown in Figure 3-10.

[2] https://www.elastic.co/products/kibana

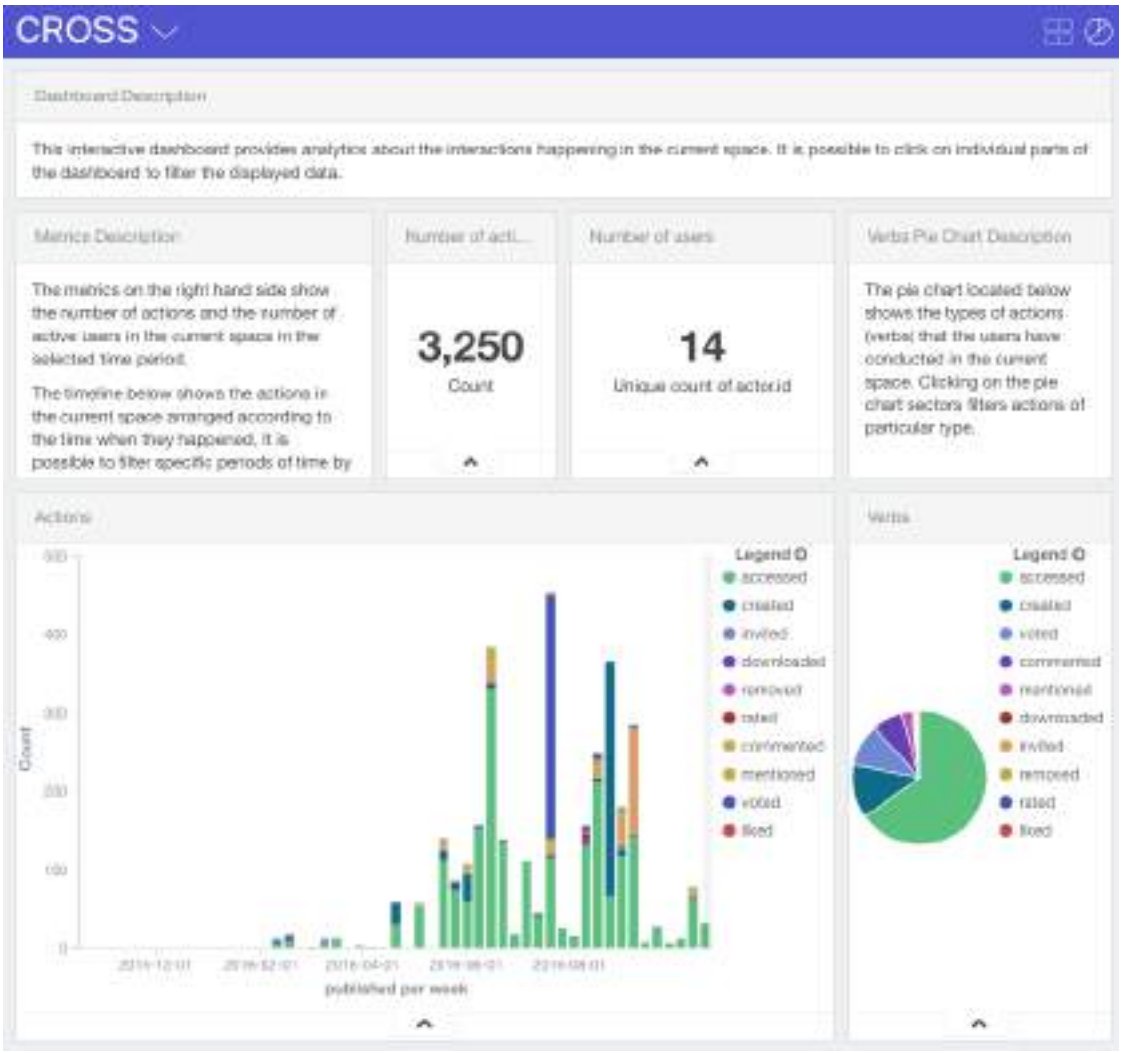

Figure 3.10. LA dashboard for spaces in Graasp.

- **IP address detection** of Graasp users and standalone users to analyse the usage per country. It allows us better understand and visualize the usage per country, a relevant aspect especially for the community cluster.
- Apps with information about whether **AngeLA** is part of the ILS or not, so that they can track or not the actions accordingly.
- A detailed description **of AngeLA** as well as of the terms and conditions to inform users (see http://graasp.eu/users/5405e202da3a95cf9050e8f9).

3.1.6 Recommendations

To support recommendations, the sustainable version of Graasp identifies potentially relevant material based on both the content and the traces generated by the user. Concretely, the following recommendations are offered:

- **Space-related**: such recommendations are provided not only based on the content but also taking into consideration the phase of the ILS. Thus, apart from a list of resources recommended because of the terms used in the space, certain apps are suggested based on their descriptions. In that way, users are supported in order to create ILSs following the IBL pedagogical approach. This functionality could be especially relevant

for newcomers since it may difficult for them to find out what are the most convenient apps to scaffold each one of the phases.

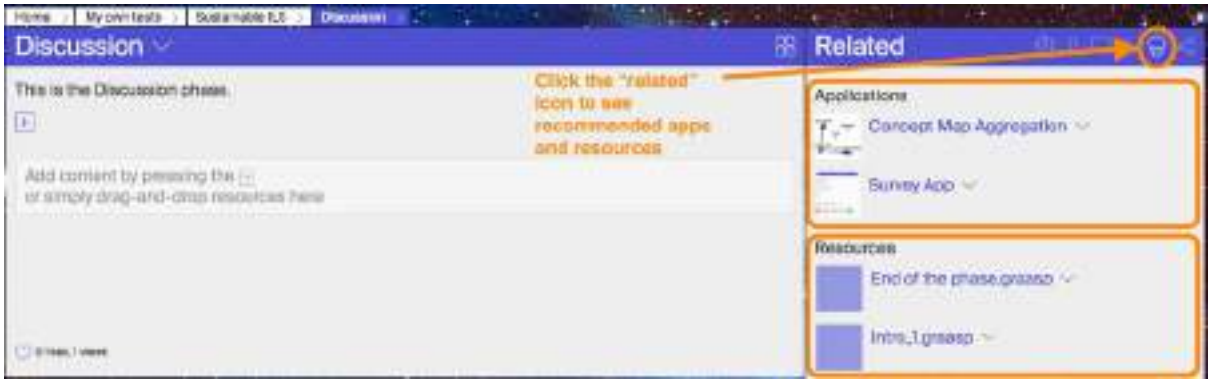

Figure 3.11. Recommendations per phase.

- **User-related**: such recommendations are offered based on the user interests. For that purpose, users are able to check the interests automatically extracted by Graasp and remove those that are not representative in their home page.

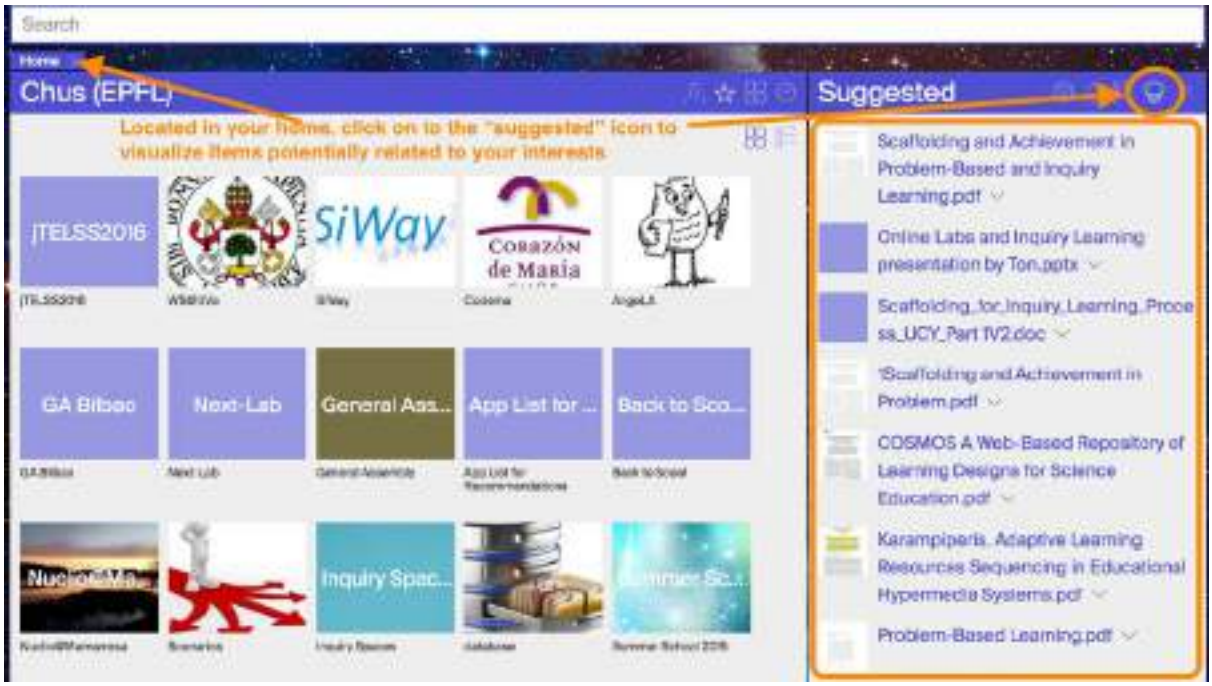

Figure 3.12. Recommendations based on the user interests.

3.1.7 User management

To support user management, the sustainable version of Graasp allows users to:

- **Invite new members**, even if they are not yet registered in Graasp, space owners will be able to send an invitation to join the space just by adding the email of the new member. The invited members will receive an email to access the platform.

Figure 3.13. Add new members by email.

- **Mention others in comments** using the mention tags such as @me, @all, @owners, @Adrian Holzer, etc. When users or user groups are mentioned in a comment, they will receive an email notification. As it is illustrated in Figure 3-14, by typing "@" in the comment, all the groups and members of the space are shown, allowing the user to type or click on the name of the relevant ones.

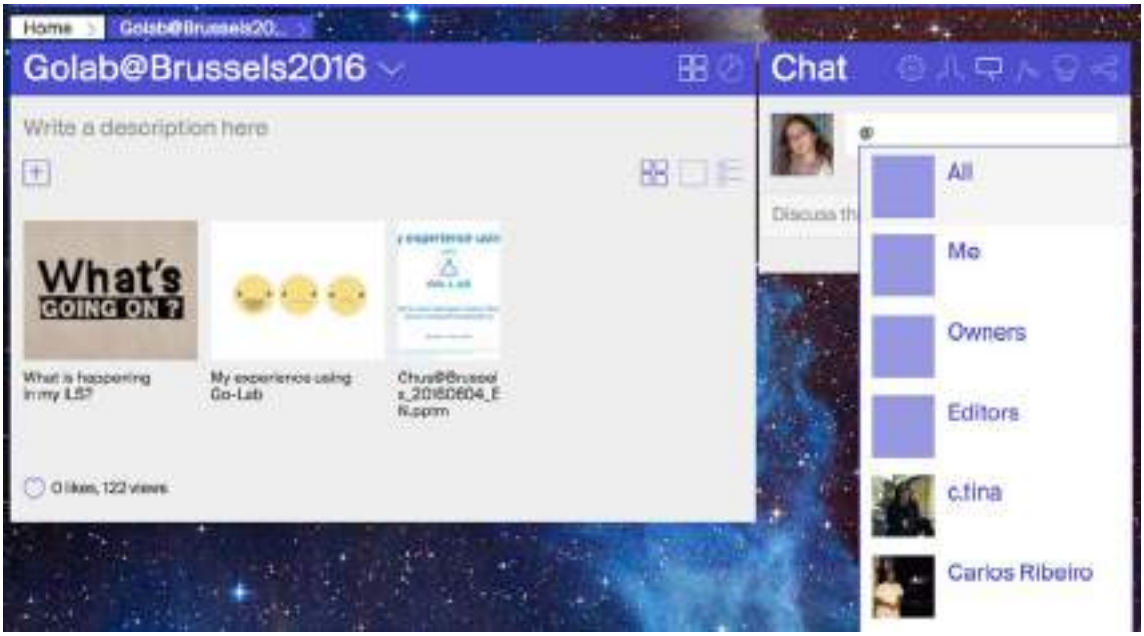

Figure 3.14. Mention others in comments.

- See the **user activity** for shared spaces on the user profile page. Reviewing the user activity of each shared space maybe too demanding in order to be aware of the last changes. Thus, this summary of the activity is expected to increase the awareness about the work done by other co-authors.

Figure 3.15. Checking the user activity for shared spaces.

- **Report inappropriate content**. In order to empower users, Graasp provides a report inappropriate content, so that any members of a space, including viewers, can report content as inappropriate. As it shown in Figure 3-16, the functionality is available when opening an item in the light box. Clicking on the "Report inappropriate content" text, a pop up appears where the user can justify the reason. Whenever a user reports inappropriate content, the owner of the space is notified.

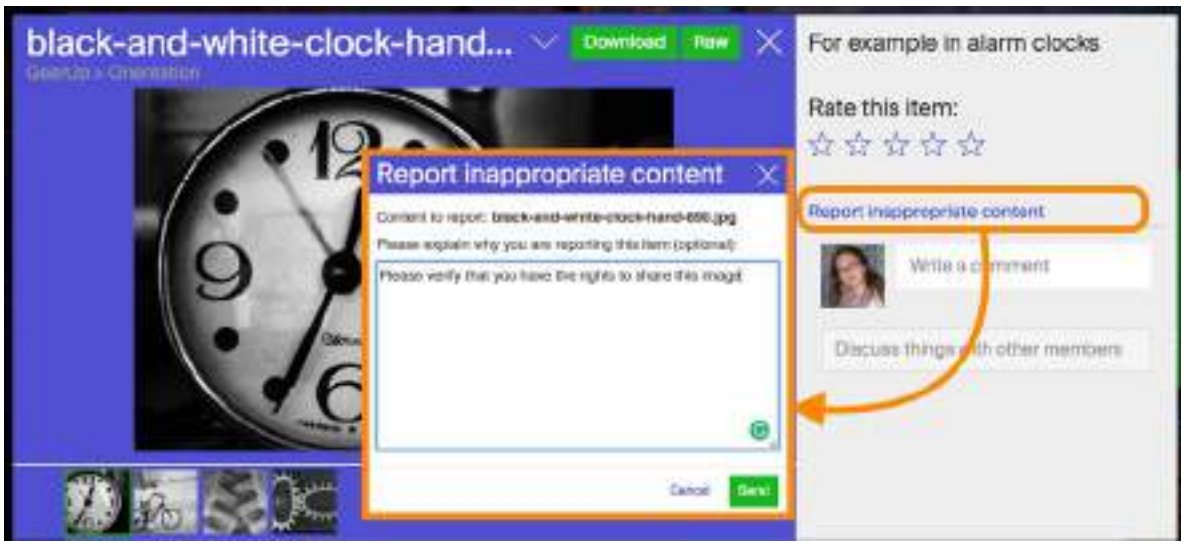

Figure 3.16. Report inappropriate content.

- **Adjust their interests**. In order to personalise the recommendations and elaborate a more accurate profile of the users, they can see the list of interests automatically extracted and remove all those that are not considered as relevant. As it is shown in Figure 3-17, to access this information, the user should be located in his/her home

space, and then open the "Interests" tab available in the right panel. Each interest can be removed by clicking on the cross located next to it.

Figure 3.17. How to visualize and customize user interests.

3.2 Non-functional requirements

3.2.1 Usability

To support usability, the sustainable version of Graasp provides:

- A **help space** with several resources to help teachers in the usage of the platform and answer their doubts either via ready to use material (such as tutorials, courses, or frequently asked questions), or by contacting experts by means of the community forum or in the tutoring platform. In this space, users can also find a link to collect their ideas about how to improve the platform. A screenshot of this space is presented in Figure 3-19.

Figure 3.19. New help space in Graasp.

- More **explicit warnings** to increase the awareness of the impact that some actions may have (e.g., removing spaces shared with other users, removing AngeLA, or deleting the Vault in an ILS). Such warnings are expected to alert users about the secondary effects that these actions may have, as well as some alternative solutions (e.g., leaving instead of removing spaces or creating a new Vault). For example, Figure 3-20 shows the warning message presented to the user when (s)he tries to delete a Vault space in an ILS.

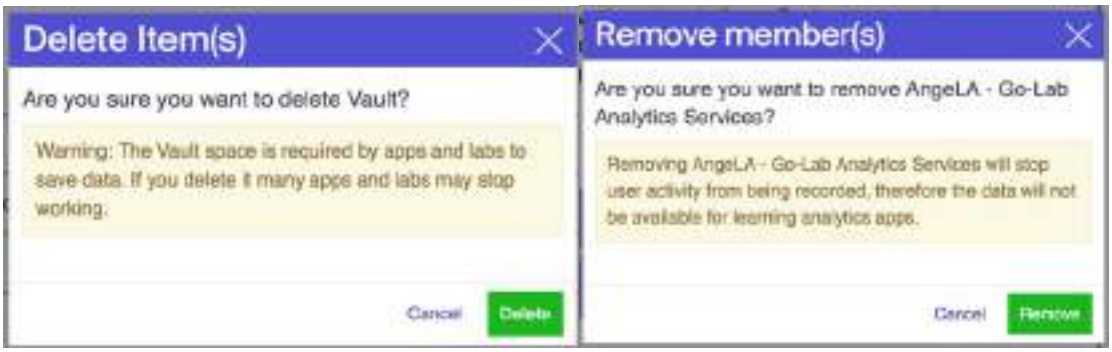

Figure 3.20. Warning message when deleting a Vault space (left) and AngeLA (right).

- An alternative **breadcrumb navigation** to the current pyramid-style navigation. The navigation style can be changed in the user settings as it is shown in Figure 3-21. Figure 3-22 exemplifies how the pyramid and the breadcrumb-style navigation for a given space. In the pyramid, each row represents a hierarchy level, where all the sibling spaces are shown. The spaces in blue represent the current space and the parents. In the breadcrumb navigation, only the path to the current space is shown. If the use clicks in one specific level, the sibling spaces will be displayed.

Figure 3.21. Navigation settings.

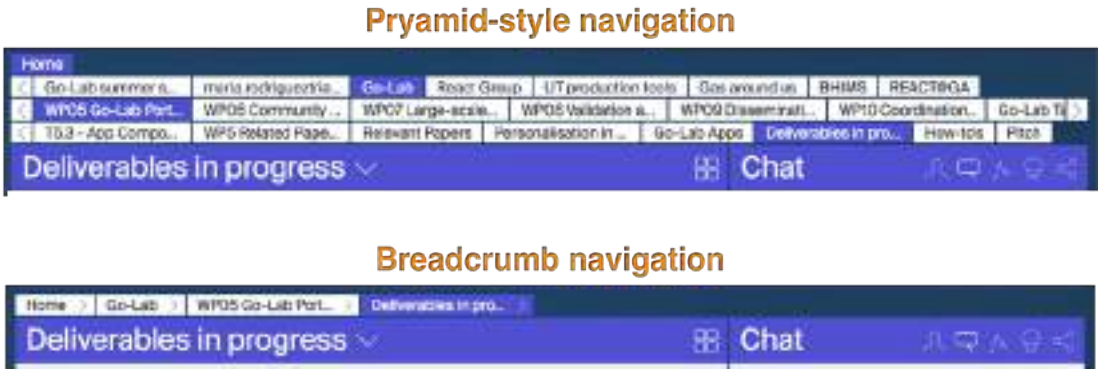

Figure 3.22. Example of visualization for the pyramid and the breadcrumb navigation.

- A **list view** of the shared spaces, enabling users to find spaces based on name, modification date, creator, visibility and privacy (see Figure 3-23).

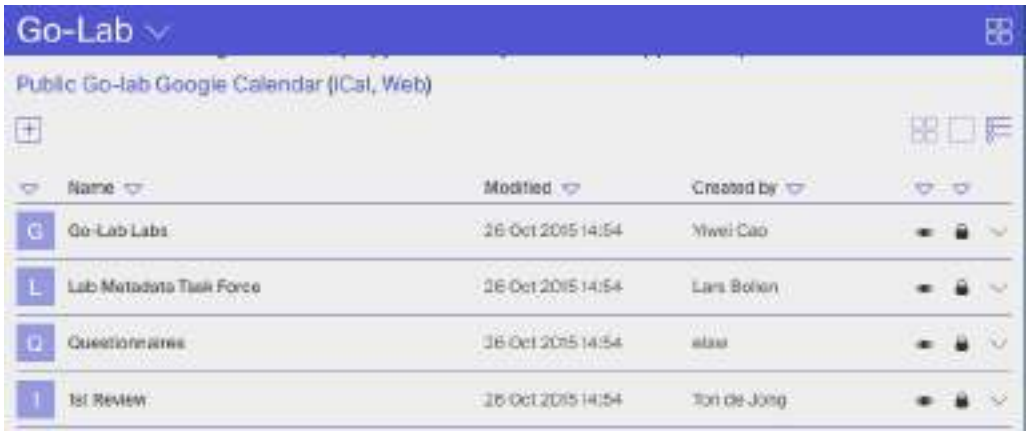

Figure 3.23. List view of the shared spaces.

3.2.2 Scalability

In terms of scalability, we have optimized Graasp and the ILS library, reducing the number of calls between the apps and the platform, as well as the amount of information sent. Concretely, we have:

- Merged the requests sent from the apps to Graasp via the ILS library to decrease the time required for initialising apps and therefore downloading ILSs.
- Allowed applications to query the Vault space using an array of values

- Offered **pagination mechanisms** in the different views (e.g., 20 items in recently visited). Since there may be many items in the same space, this improvement reduces loading time.

Apart from that, in order to reduce the computation time and thus increase the number of simultaneous users, we have adapted Graasp so that it can be run on multiple cores.

3.2.3 Internationalisation

To support internationalisation, the sustainable version of Graasp has been translated into the languages of the different partners, namely Dutch, Estonian, French, German, Greek, Portuguese, and Spanish, as well as Bulgarian, Chinese, Hungarian, and Russian thanks to the collaborations established with the teachers involved in these countries. We expect that the availability of such languages promote the adoption and facilitate the access to new European teachers and beyond. In addition, we have created a **translation mechanism** via the app composer (see Chapter 4 for more details) to allow users easily translate the platform to their language if it is not yet available.

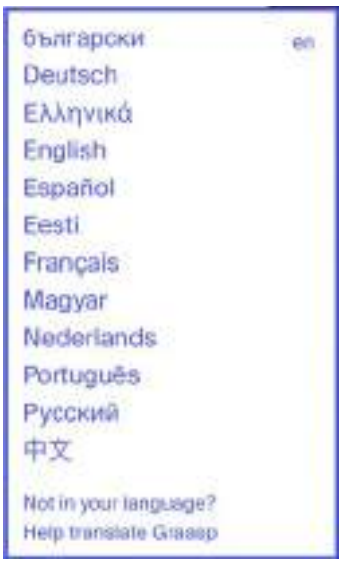

Figure 3.24. Language menu with the languages available and the link to the translation mechanism.

3.3 Summary

As a conclusion, the aforementioned features have a threefold purpose:

- First, some of these features aim to **satisfy user needs** that have been proposed by the users themselves (e.g., creation of discussion threads and hints, sharing ILSs via QR codes, and generating printable content).
- A second set of features aims to improve the **user experience** by offering user support (such as introductory material and a help space), improving usability (providing warnings to prevent users from undesired actions, offering alternative navigation schemas, and decreasing download time), increasing awareness (via contextual analytics dashboards and activity notifications), and facilitating content discovery (by means of the search and recommendation mechanisms).

- Third, part of these features aim to facilitate the **community building** by increasing the interaction between users (by means of the extended communication channels and helping them discover users with similar interests), promoting the sharing and reuse of learning content (thanks to the republishing and search mechanism), inviting new users, and allowing users to translate the platform to any language they need.

4. The App Composer

This chapter details the functional and non-functional requirements of the App Composer. The App Composer is a tool that provides two components in its final form: App Composer Translator (for assisting teachers and lab owners in translating apps) and App Composer Embed (for integrating external non-standard apps in the Go-Lab repository). In the past there was a component called App Composer Adapt for adapting applications. But this type of adaptations is now made through the Apps themselves and stored in the ILS platform. Therefore, as detailed in D5.6, the App Composer Adapt is still supported but in a legacy mode: no new adaptations should be added through it, but those existing still work.

Therefore, the App Composer does not target students nor teachers. The only exception is the App Composer Translator, but only certain invited teachers can translate apps to their native language. The majority of teachers just see that apps are translated, without requiring them to be aware of this component. Given this, we describe these requirements in the “lab owners and app creators” section.

4.1 Functional requirements

4.1.1 Publishing non OpenSocial apps in the repository

The App Composer Embed supports the inclusion of external resources in the Go-Lab repository as if they were OpenSocial apps. The lab owner / app creator can include different resources in different languages, and an OpenSocial app is generated and hosted by the App Composer Embed. When teachers/students access this app from an ILS in one of the languages provided, they will see the app in that language.

4.1.2 Translation of apps

App creators using the OpenSocial standard will be able to use the App Composer Translator or send the link of the App Composer Translator to people speaking different languages. Given that the App Composer Translator relies on Cloud services for suggesting texts (such as Microsoft Translator API and Google Translate), translators are able to translate texts faster than if they had to deal with them manually. Furthermore, messages translated are applied automatically in the ILS Platform without requiring any action from the lab owner or app creator.

Lab owners not using the OpenSocial standard but using the Smart Gateway to include their laboratories can use the Smart Gateway (see D4.7, Section 3.4.4) to export their messages in an automatic way. This way, the App Composer Translator can take these messages and show them to translators. However, in this case, the translations are not automatically available in the ILS Platform, requiring some action from the lab owner / app creator. For this, the App Composer provides a notification mechanism that sends e-mails to them to notify them that apps have been translated.

4.1.3 Translators finding apps to be translated

The App Composer Translator provides a list of all the apps and labs available in the Go-Lab Portal which can be translated. Therefore, translators of one application will find the rest of the apps and labs, and might translate them too. This way, translators of one app can easily spot and translate other apps and labs.

Furthermore, it becomes easy for translators to find those apps or labs not translated to a particular language, as seen in the following figure for Spanish:

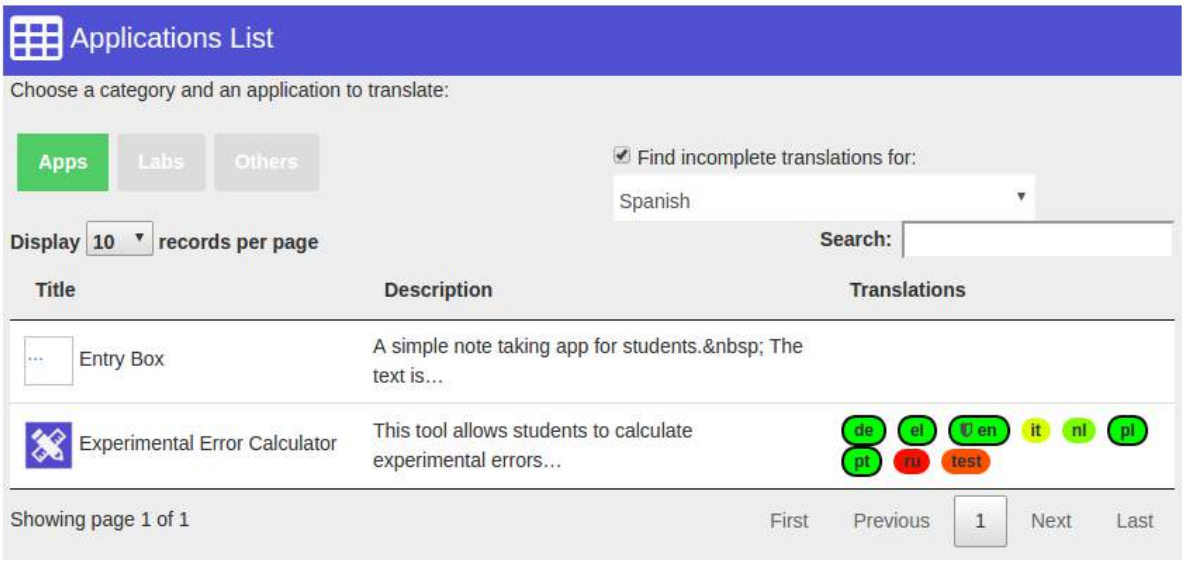

Figure 4.1. Application List in the Translator.

4.1.4 Automatic update of the language metadata in the Go-Lab Portal

The App Composer Translator provides a service detailing what apps and labs are available in what language. The Go-Lab Portal automatically retrieves this information periodically and uses it to update the metadata of the applications. Therefore, if a teacher translates an app to a language, in a matter of hours, the Go-Lab Portal will report that the app is supported in that language in a seamless way without requiring further action by the app creators, lab owners or administrators of any service.

4.2 Non-functional requirements

4.2.1 Scalability

The App Composer should be able to serve thousands of teachers, so scalability is fundamental for providing a proper user experience.

In the App Composer Translator, all the translations are stored in an intermediate cloud database, so there is no connection from the users rendering applications to the App Composer. The App Composer is only used when a teacher is actually translating an app.

Therefore, none of the App Composers is expected to have a high number of connections or scalability problems.

4.2.2 Translation Maintenance

The App Composer provides a set of tools and stats pages, listed in:

- http://composer.golabz.eu/translator/dev/
- http://composer.golabz.eu/translator/stats/

These pages display further information about the current usage of the App Composer. These include:

- A web page[3] displaying all the apps not using OpenSocial which have been translated by human translators but that the lab owners have not applied those translations. For example, a lab owner not using OpenSocial (e.g., using the Smart Gateway) may provide the interface in 3 languages, and still export the messages, and a teacher might have translated the user interface to a fourth language. The lab owner will be notified whenever this happens, but maybe the lab owner fails to update the text or forgets about it. This page provides Go-Lab administrators the chance to see who has not yet applied the translation and contact them as a reminder.
- A web page[4] showing an overview of the translation status per language: displaying what apps are missing per language. Go-Lab administrators can use it in the future to contact particular translators and request them to translate missing apps, or apps that have been updated and now contain new texts.
- A web page[5] showing the apps and all the potential files that can be downloaded by lab owners or app creators to install those translations in their servers. It contains a revision panel (for example[6]), where administrators can see who translated what and when.
- A web page[7] showing how many apps have been integrated in the Go-Lab Portal using the App Composer or the Smart Gateway.
- A web page[8] listing the users using the system and the number of contributions.
- A web page[9] showing the current translations pushed to the database located in the ILS platform. This is useful for confirming that the integration between the App Composer and the ILS platform is working properly.

In addition to this, a notification mechanism to tell existing translators about new contents to be translated is being implemented. When an app is updated and has new messages, or a new app is added to the repository, the App Composer notices that there are new messages to be translated to different languages. The new feature will consist in that weekly an automated email will be sent to translators who have already used the tool to translate apps, notifying them about this new content, so they can go to the tool and translate it.

4.3 Summary

As a conclusion, the features listed above have the following purposes:

- Reducing actions required by humans, increasing the level of automation and reducing the number of integrity errors. For instance, whenever a message is translated, the Go-Lab portal is automatically updated both for apps and labs, and subsequent searches are accordingly applying the new results.

---

3 http://composer.golabz.eu/translator/stats/missing
4 http://composer.golabz.eu/translator/stats/status
5 http://composer.golabz.eu/translator/dev/apps/
6 http://composer.golabz.eu/translator/dev/apps/revisions/it_ALL/ALL/http://go-lab.gw.utwente.nl/production/edt3/build/edt.xml
7 http://composer.golabz.eu/translator/stats/status
8 http://composer.golabz.eu/translator/stats/users
9 http://composer.golabz.eu/translator/dev/mongodb/apps/

- Encouraging participation: by a) quickly showing translators other apps to be translated, b) showing administrators statistics on what app translations are missing or what languages are better supported so as to contact translators, and c) providing support for non-OpenSocial apps with the target of attracting more translators that potentially could translate other Go-Lab apps.
- Reducing maintenance costs by supporting a scalable, asynchronous approach as well as by reducing activities required by humans (as previously noted). A problem in the App Composer will not make the rest of the tools slower and will only affect new translations or updates; not existing translations.

5. Inquiry Learning and Learning Analytics Apps

This section presents the latest developments in inquiry learning and learning analytic apps towards a sustainable set of scaffolds to support learners during their work in ILSs. In addition to what has been described in previous deliverables (particularly in D5.5 “Release of personalisation features and inquiry learning apps - final”), the latest developments are described in the following sub-sections. The developments aimed at creating a complete and final set of tools, by solving open issues, and adding missing features.

5.1 Latest App Developments

5.1.1 Reporting Tool

URL on the lab repository: http://www.golabz.eu/apps/report-tool
Supported languages: English
Predominant phase(s): Discussion, Reflection

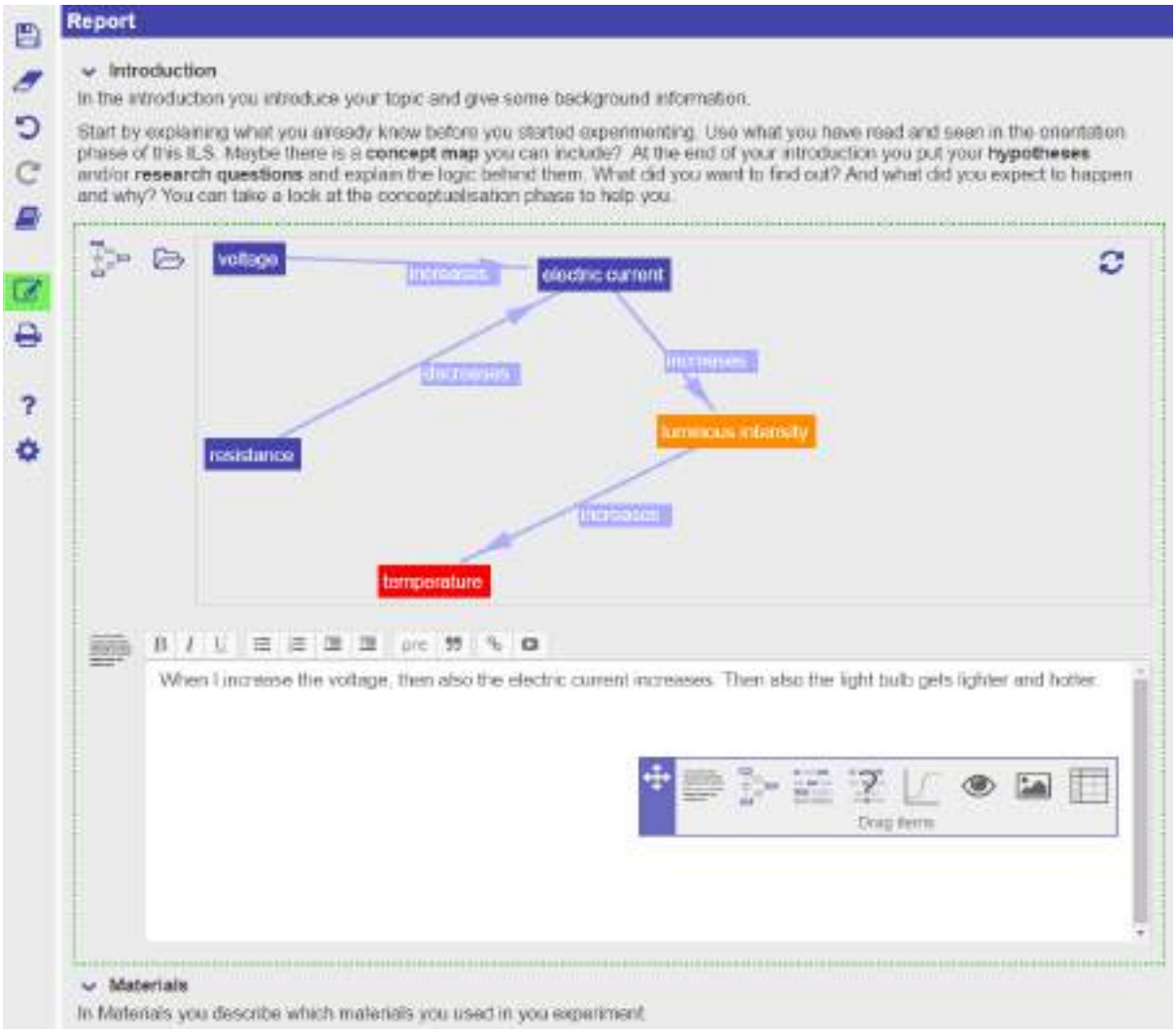

Figure 5.1. Screenshot of the Report Tool with imported concept map.

Short description:
The Report Tool aims at supporting learners to compile a report from their experiences towards the end of an ILS, e.g. in a discussion or reflection phase. Learners can reuse products from earlier phases, such as concept maps, hypotheses, questions, observations and data graphs, and enter text to describe their findings.

Description:
The Report Tool supports learners to create a final report from their inquiry learning experiences. A report is typically structured much like a research article, with an introduction, description of the experiment design, data analysis and conclusion sections. A teacher can configure a report by defining the number of sections, the titles of sections, and by adding a short descriptive text to each section that contains instructions about what and how to fill in a section.
In each section, a learner can import artefacts from other phases and tools, like concept maps, hypotheses, questions, observations and data graphs. The reuse of resources is supposed to help learners to create a coherent and consistent report about their activities, while understanding the iterative research process from understanding a topic and learn about the concepts and relations in a domain, to the design and realisation of experiments, up to the analysis and evaluation of data and to report and discuss the findings.

5.1.2 Survey App

URL: http://www.golabz.eu/apps/survey-app
Available languages: English, German, Italian, Spanish, Polish, Dutch, Hungarian, Portuguese, Greek, Russian
Predominant phase(s): Discussion, Reflection

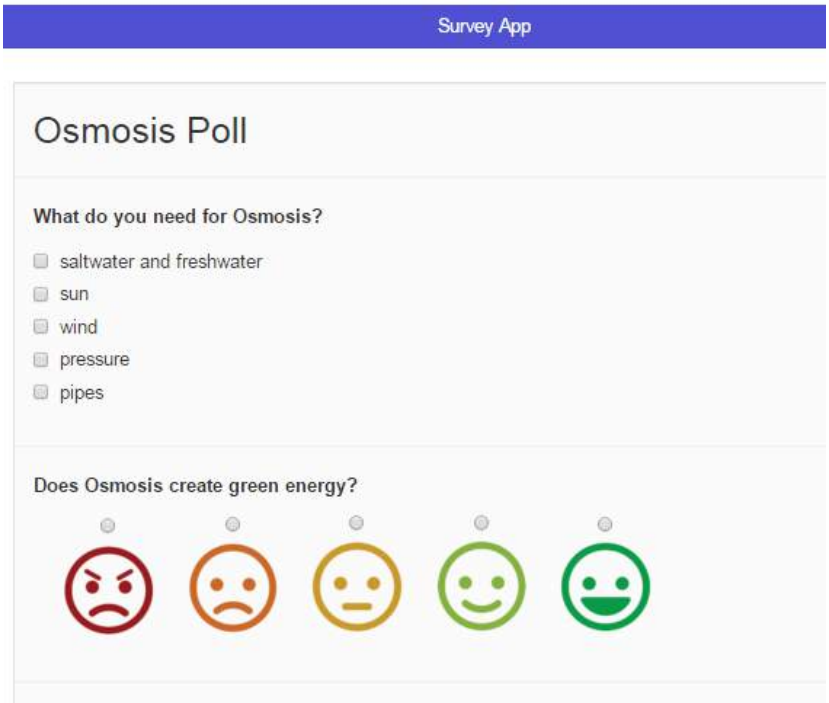

Figure 5.2. This survey has been created with the app. It contains a multiple choice and a single choice question with results on a Likert-scale, underlined by emoticons.

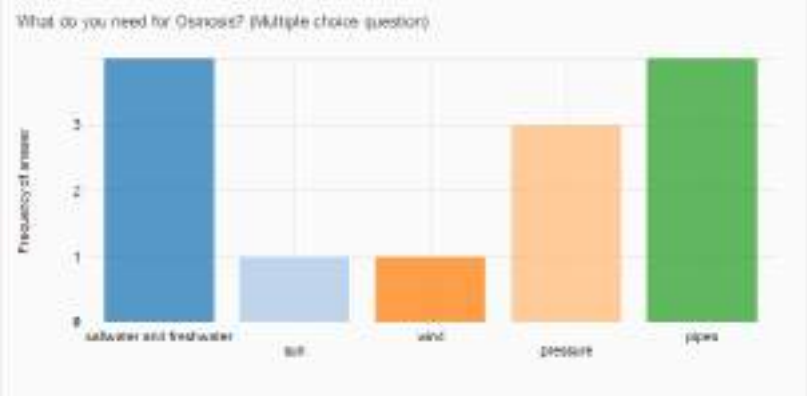

Figure 5.3. The responses of multiple choice questions are displayed in a bar chart in the teacher view.

Figure 5.4. Results of open text questions are displayed one-by-one in quotation marks.

Short description:
The Survey App helps teachers to create surveys and distribute them towards their students. The student responses are then collected anonymously and are visualized in the teacher view.

Description:
The Survey App enables teachers to build your own surveys, questionnaires, and polls. These forms are published through an Inquiry Learning Space so that the users of such ILS can respond to the questions. The results are collected anonymously so that it can be used for evaluations or for reflection. The results can be displayed in the teacher view of the app, for each question with a visualization corresponding to its type. During the creation of surveys, different types of questions (single choice, multiple choice, etc.) are available. Questions and possible answers can be authored by the teacher regarding the chosen type of question. The created survey has an external and portable format, so every survey can be distributed and copied to another ILS. Certain pre-defined templates, such as questionnaires for self-assessment of performance of emotions exist. The survey app uses the learning analytics server for the anonymization and aggregation of the responses in order to prevent the teacher from seeing the original results.

5.1.3 Concept Map Dashboard

URL: http://www.golabz.eu/apps/concept-map-dashboard
Available languages: English, German, Italian, Dutch, Russian
Predominant phase(s): - (ex post usage, teacher only)

Short description:
The “Concept Map Dashboard” supports the reviewing of student concept maps, giving insights in group knowledge and its development over time.

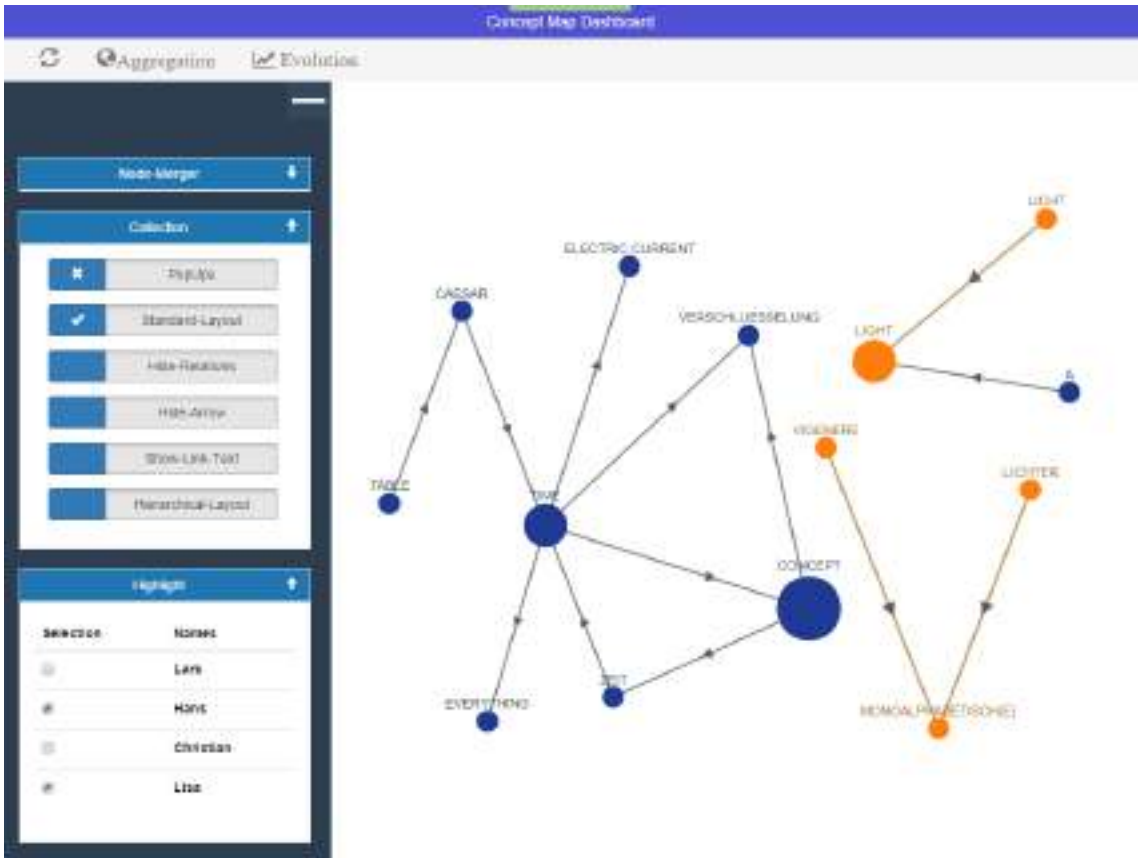

Figure 5.5. In this aggregated concept map, the concept maps of two students are highlighted in orange colour.

Figure 5.6. The evolution view displays several measures over time: in this example, it shows node count over time for six students.

Description:
The “Concept Map Dashboard” consists of two parts: a) CM Aggregation, and b) CM Evolution. The aggregation is similar to what the Concept Map Aggregation (Deliverable D4.8) does, it creates an aggregated graph as a union of all student concept maps. This app features an improved version, which enables interactive filtering and highlighting of individual concept maps. Additionally, it allows to define synonyms for concepts graphically, which then influences the aggregated results (as synonyms are aggregated into the same concept). The evolution shows the development of student concept maps over time. It features several metrics such as node coverage corresponding to the aggregated map or node count as metrics to be displayed in a chart over time. It uses the services of the learning analytics server for the aggregation and evolution.

5.1.4 Group Formation App

URL: http://www.golabz.eu/apps/semantic-group-formation-app
Available languages: English, German, Russian
Predominant phase(s): - (ex post usage, teacher only)

Short description:
The “Group Formation App” creates heterogeneous learning groups based on diverse knowledge.

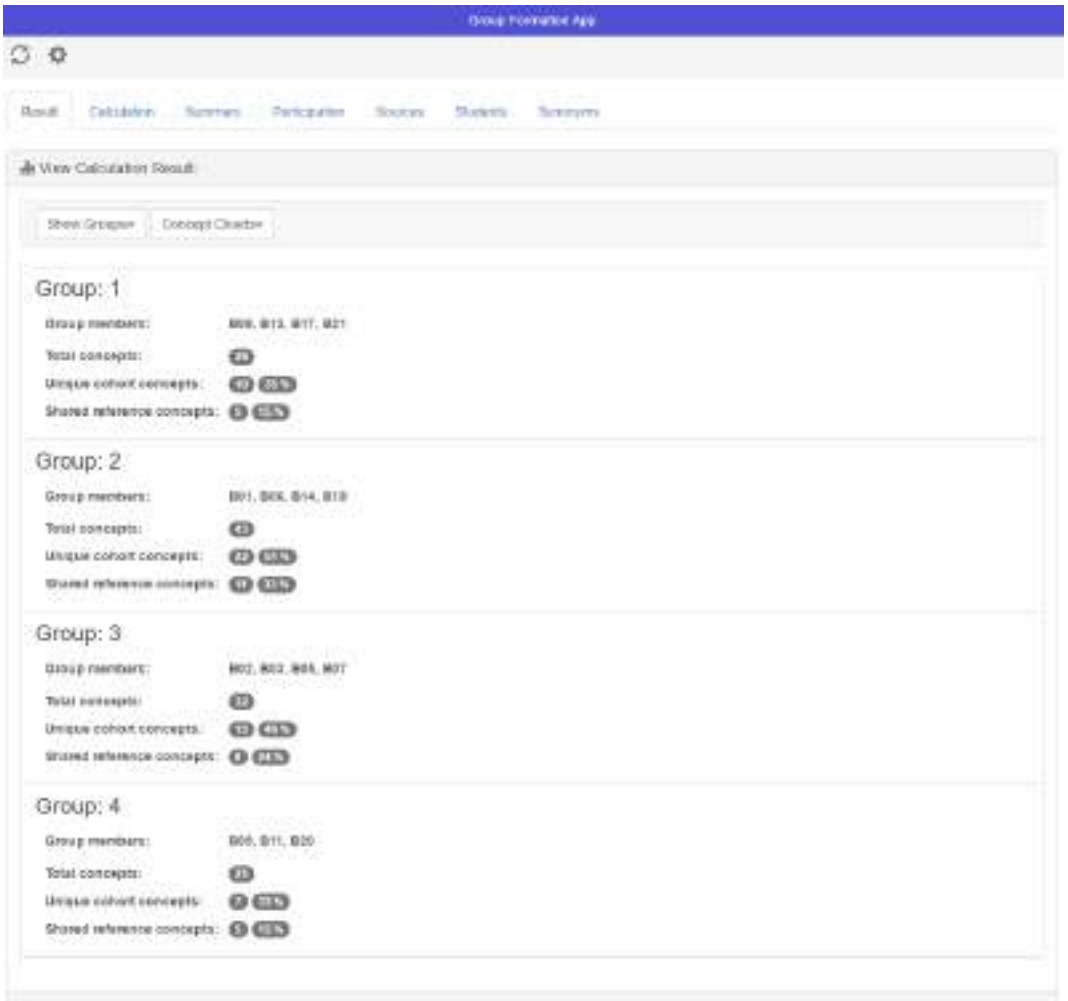

Figure 5-7. The results of the group formation includes metrics such as the number of shared or unique concepts per group.

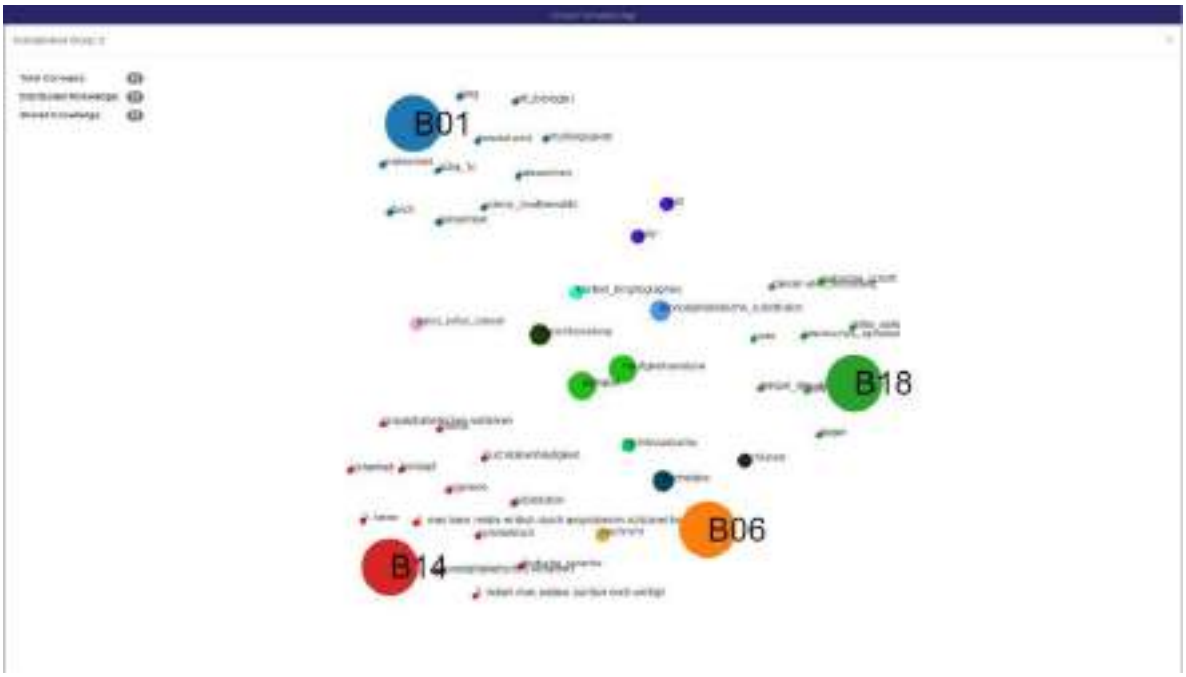

Figure 5.8. The teacher can get an overview about the group knowledge in the visualization above. It presents unique concepts and shared concepts in a way similar to Venn diagrams (but without the circles, only implicitly through colour schemes).

Description:
The idea behind this app is “semantic group formation”. Heterogeneity refers to diverse knowledge in this sense. Students create artefacts (texts/wikis, concept maps, hypotheses), which represent inherent domain knowledge corresponding to certain terminology. These concepts are automatically extracted from the learner-generated artefacts in the vault using the same data model and service as the concept cloud. A group formation is calculated based on this model. It optimizes the semantic diversity among all learning groups.
Additionally, the teacher can use this app to see graphical representations of the group knowledge and the diversity measures. The app provides more measures and indicators to present both the outcome of the group formation and the function of the algorithm to the teacher.

5.2 Code repositories

The accessibility to the code, documentation and licenses is a key element in order to facilitate the maintenance and sustainability of apps, labs and libraries. Thus, as it was already done for the libraries (https://github.com/go-lab/ils), the code corresponding to apps and labs is available in the following repository: https://github.com/go-lab/apps.

Regarding licenses, ILS common libraries, all smart-device templates, and smart-device-metadata are shared under MIT open source license. The same applies for iwc (enables the interwidget communication and drag and drop between widgets). The apps and labs developed within the project framework are shared under CC BY-NC license[10].

5.3 Summary

Concerning the apps and supportive scaffolds, the development of the last year include:

- Improvements in usability, stability, performance, and responsiveness.
- New scaffolding app releases such as the Survey app and the Report tool, which has been only been only in a prototype state until now, has been finalized and released to the Go-Lab portal.
- New learning analytics tool releases that help interpret and understand the students’ results and artefacts, such as the Group formation app, the concept map dashboard.
- Provide graceful degradation of functionalities and notification of problems (e.g., lack of Vault, AngeLA on/off, backend up/down, bad/good network).

[10] https://creativecommons.org/licenses/by-nc/2.0/

6. Conclusion

In this final deliverable of WP5, we have highlighted the changes we have made to the different Go-Lab solutions in Year 4. These changes focused on the sustainability of the Go-Lab platforms to enable their long-term use.

In Golabz, the main changes involved improving the usability for lab owners to encourage them to publish labs, adding quality control mechanisms to ensure the content of Golabz is relevant for users, and improving support to make it easier for teachers to create, (re)use, and share ILSs.

In Graasp, the main changes involved empowering teachers by providing more awareness and help, improving usability to reduce technological friction to creating ILSs, and encouraging collaboration with other teachers to contribute to the creation of a self-sustained teacher community.

The apps have also been polished in the last year to improve usability and a graceful degradation mechanism has been integrated to enable them to work even if some parts of the system are down (such as the LA backend) or some settings are turned off (AngeLA) or missing (the Vault is absent).

A sustainable internationalization of Graasp, apps and labs is now possible thanks to the App Composer. Besides, the App Composer has extended its functionalities in order to encourage app and lab owners to share them in Golabz, facilitating the adaptation process of third party apps and labs and providing statistics of their usage. In general, the app composer has been further automatized wherever possible and where it was not possible, an effort was made to make human action simpler to encourage participation.

Apart from improving the user experience in terms of functionalities and usability, a significant emphasis has been put on scalability. Given the increasing number of users joining the platform, their daily interaction with the different platforms, and the resources created by them, it was essential to provide service for large scale usage. Thus, the different Go-Lab solutions have been refined in order to improve the communication between platforms, minimize the traffic load, and reduce the response time.

It should be noted that some efforts during the first 3 years of the project already had sustainability of the platform in mind. Designing a useful and user-friendly platform was a central aspect of the project which was achieved through inter-cluster coordination and participatory design. Furthermore, from a technical perspective, the loose coupling of components allows to add and remove modules if needed without affecting the whole system. Also in terms of technology, we performed a complete redesign of Graasp in Year 2 to the latest state of the art web technology, which made it well prepared for future developments.