SuperForm: Case Description

1. Context and project definition

At UCL, we use a variety of communication channels to publish information to students, researchers, colleagues, or a broader public. The channels used depend on the purpose of the communication and the targeted public. Some communications should get distributed on multiple channels. In that case, although the main information to be distributed remains the same, each channel may have a slightly differing format or layout, depending on the medium for which it is intended.

To facility the secretaries or other personnel members to write news items and distribute them on these different channels, we decided that it would be useful to create a new application that would allow us to encode the main information once, and then distribute it on the chosen channels, optionally after having adapted the news according to the required format of each of those channels.

Once encoded, this application will then distribute the news, correctly formatted, on one or more media depending on the choice (and rights) of the person who encoded the news. Encoded news items will always need to be validated first by a moderator in the application (for example the local department secretary).

2. Goals

The primary goal of this project is to generalize the encoding and sharing of news information through a single data entry point. This saves valuable staff time.

It should also be possible to validate the data inserted by a user. For this purpose, the application will allow content moderators to be assigned, whose role it is to validate news items inserted by others, before they can get published.

A third objective is to structure the common information shared on various channels, in order to standardize the published new items.

A final objective would be to archive the different news items encoded or published earlier, in order to edit or reuse them in the future, and to avoid the double encoding of similar news items.

3. Scope

The SuperForm application should enable users to distribute news on the following channels:

- **CCII Employment**: the CCII Employment portal (http://www.ccii-emploi.be) is a web portal with job offers managed by the CCII engineering student organization.
- **Mailing lists**: at UCL and elsewhere, various e-mail mailing lists exist. To post news to the members of such a list it suffices to send a structured email to the list's email address.
- A **wiki page** used for sharing news with students and/or personnel members.
- **ICTV**: a channel editor of the information TV screens in the corridors of our institute, on which news can be posted via JSON.
- **RSS feeds**.
- A **Facebook account** of the department or institute, where information is shared with students, researchers or a broader public.
- A departmental **LinkedIn** account or group: LinkedIn (https://www.linkedin.com/) is a social network service comparable to Facebook but mainly used for professional networking.
- **Google+**: another internet-based social network, that is operated by Google.
- **Twitter**: shorter news messages of a limited amount of characters only.
- **Gcal**: to automatically publish the dates of upcoming events in a shared google calendar.

For each of these channels we can limit ourselves to channels that are reserved for internal usage at UCL. In other words, the application will only be accessible by users connected to the UCL network.

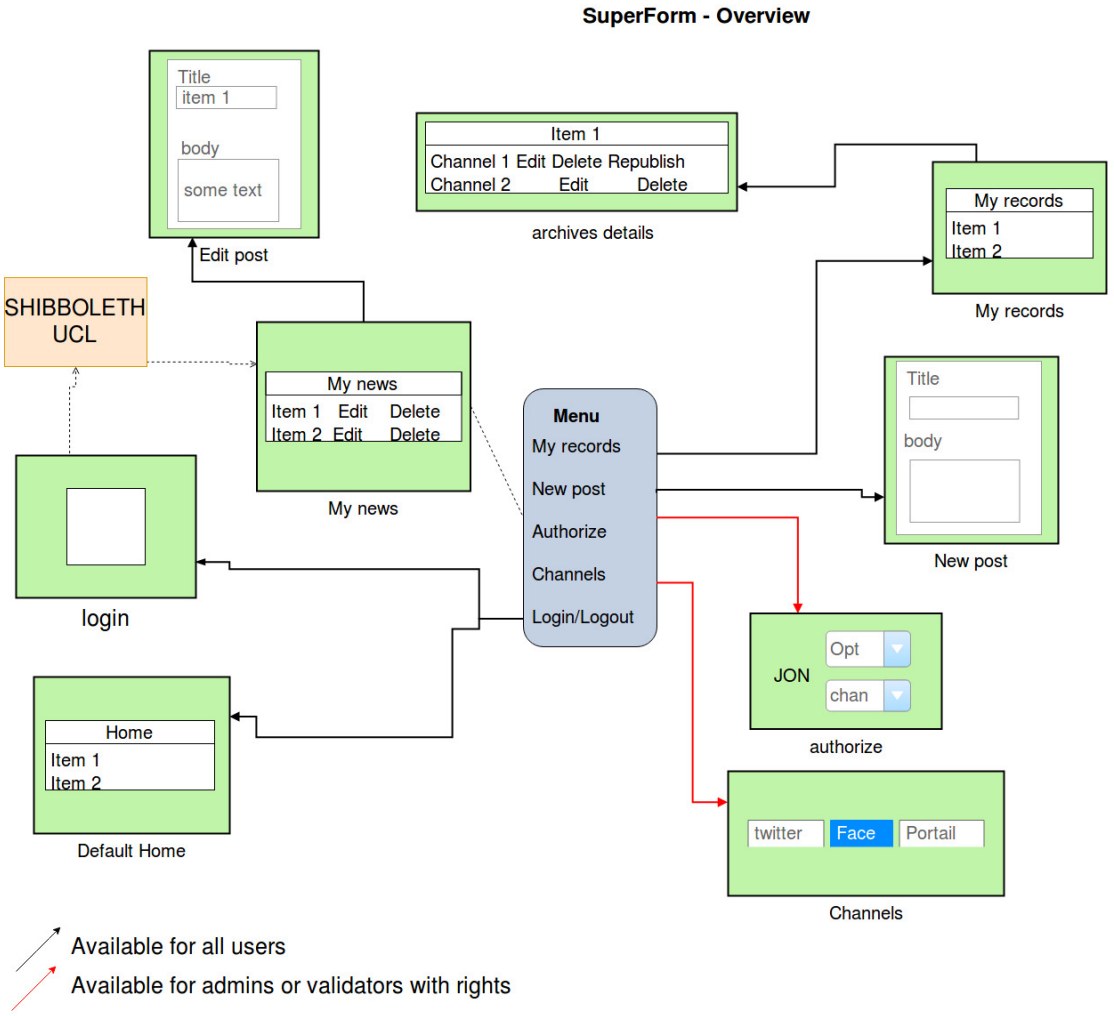
4. Functional requirements

4.1 Existing application

Before the start of this project, the client will already have developed a small open source prototype of the application that allows publishing (and moderating) posts by e-mail only. The existence of this prototype imposes certain analysis and implementation choices that will be explained below.

4.1.1 Structure of the application

The schema below summarizes the overall structure of the application. In what follows, each of the functionalities shown in this schema will be explained in more detail. We refer to Chapter 7. Appendix : User Interface Screenshots, for a more complete rendering of the user interfaces of each of these components.



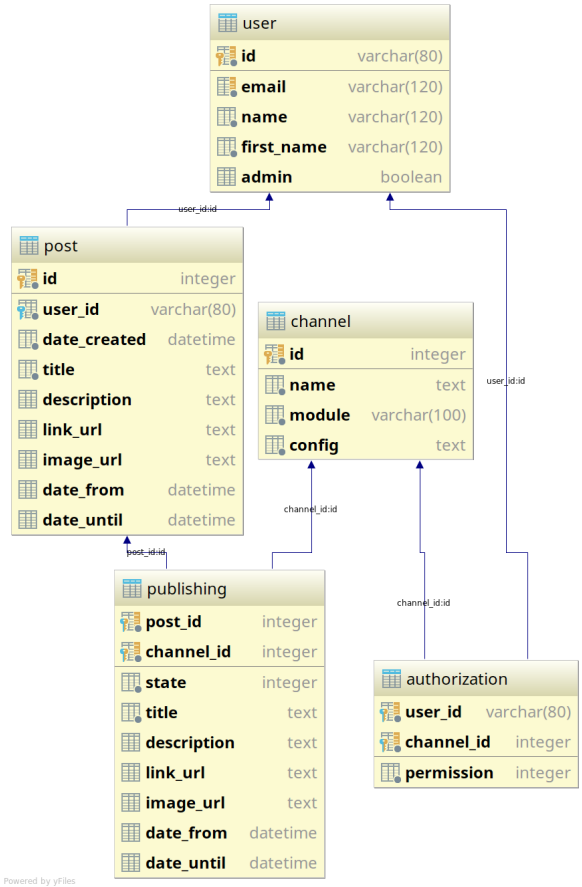
4.1.2 Shibboleth

Please note that the application is connected to the SSO (Shibboleth) of the UCL. This allows to directly connect to the UCL structure and simplify the management of users.

What is Shibboleth? Shibboleth is an open-source project that provides Single Sign-On capabilities and allows sites to make informed authorization decisions for individual access of protected online resources in a privacy-preserving manner (cfr. https://www.shibboleth.net/).

An explanation to let you use a SSO system during your development will be provided later in a separate document.

4.1.2 Initial database

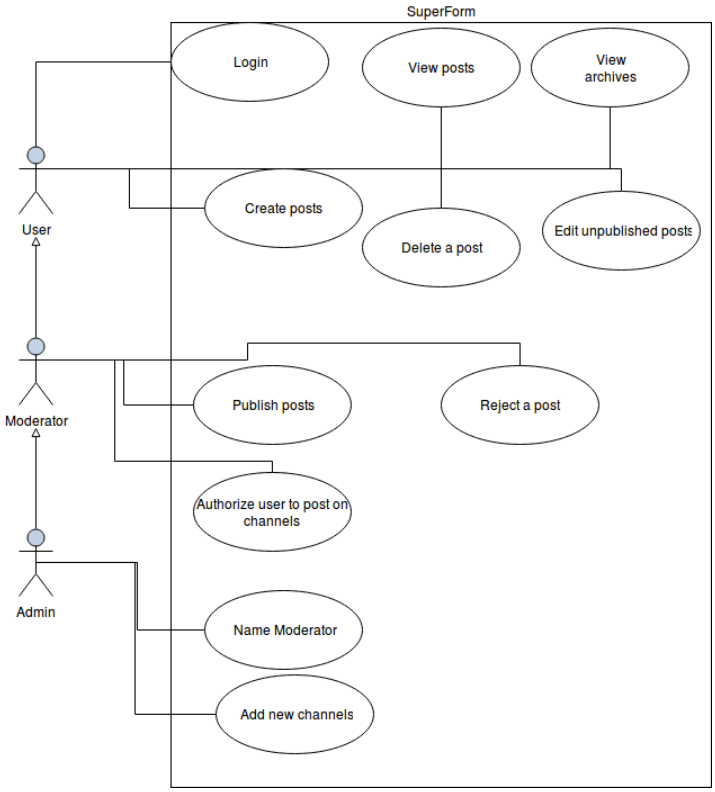


The application's database schema has a fairly simple structure. *Users* can create different posts of news items. These posts can be published on different channels. What channels a user can publish on is determined by the authorizations that have been given to that user.

- User, posts and channels are uniquely identified by an id.
- Whether a user is an administrator or not is indicated by a Boolean field admin.
- There is a uniqueness constraint on the user email. (Given that the application will be reserved to UCL users, we require them to use their UCL email address only.)
- Publications and authorizations are identified by a primary key combination: the post id and channel id for a publication; the user id and channel id for an authorization.
- A post represents the general information about a news item to be posted, whereas a publication corresponds to an instantiation of that post for a particular communication channel. The duplication of information between publications and posts is necessary because

a publication can adapt the general information of a post to the specific needs of the channel on which the information is published. (For example: the Twitter channel limits the text description of a published news item to max. 320 characters.)

- The status of a publication is represented by an integer. The possible states are: -1 = incomplete, 0 = not validated, 1 = validated/shared and 2 = archived. More information on these states and their life cycle will be given in the next subsection.
- Similarly, the permission level of a user is represented by an integer. For the moment we have only two permission levels: 1 = author, 2 = moderator. Moderator access subsumes author access, in other words a moderator can also do everything what an author can do.
- Note that a user can both be author and moderator depending on the channel he has access to. He can be author for one channel but moderator for another.
- As illustrated below, three different kinds of users with access to increasingly powerful application features are distinguished: (normal) users, moderators and administrators.

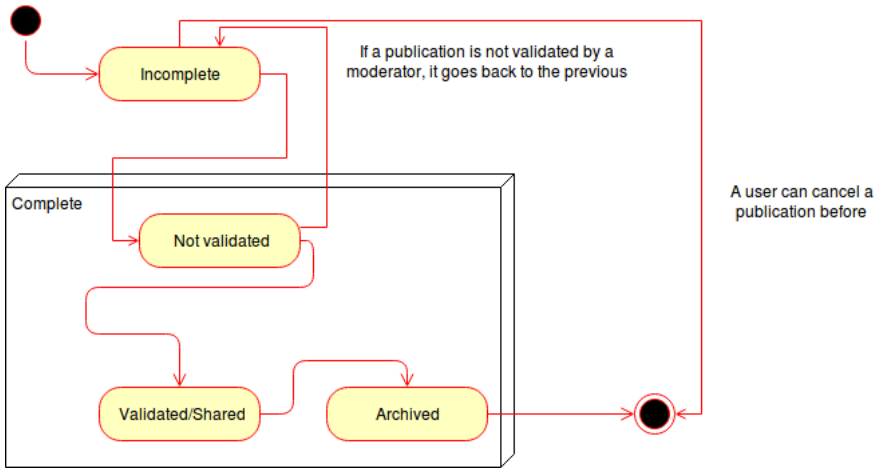


In this use case diagram, the arrows between two different roles should be read as: “I can do the same things as my parent role but I also have the additional features shown to my right”.

- A configuration describes the properties of a particular channel. It is a dictionary (key,value) of data that the channel will need, in order to be able to post information on that channel. In the database, configurations are encoded as JSON descriptions. For example, for the channel of the INGI mailing list, the configuration will be a dictionary like:

```
{ prefix : "[COM INGI]",
  mailing_list : "communaute-ingi@listes.uclouvain.be"
}
```

4.1.3 Life cycle of a publication



The state diagram above describes the different states that a publication can be in, in the current version of the SuperForm application prototype. After its initial *creation*, as long as a publication has not been submitted for validation, it is in an *incomplete* state, and can still be edited and completed by the user who created it. Once submitted for validation, the publication is in the *not validated* state, where it awaits validation by a moderator. If the moderator validates it, it goes to the *validated/shared* state and gets shared on the indicated channel; if not it goes back to the *incomplete* state, where the user can decide to edit it again and resubmit it for validation, or destroy it if he decides the news is no longer relevant. Finally, once the publication date has expired, the publication will go from *shared* to *archived* where it remains for future consultation or until the archive gets emptied.

This life cycle is followed individually by each publication (i.e. each particular channel) corresponding to a post, even if a user would want to post a same news item to different channels at once.

4.2 Requested application features

Starting from the existing prototype application as a basis, we call upon you to complete it with different additional features, as explained in this section.

4.2.1 Creation of different channels

As mentioned above, different channels need to be added for publishing news items. As an example for you from which you can take inspiration, the prototype application already provides an implementation for the e-mail communication channel. In addition to this, the following channels will need to be added:

1. CCII Employment
2. Wiki
3. ICTV (a channel editor) via a JSON
4. RSS feed
5. a Facebook account
6. a LinkedIn account or group
7. a Google+ account or group
8. a Twitter account
9. a shared Gcal

Note that these different channels do not all have the same data needs. So be alert and intelligent in your selection of what information should be published on what channel and in what format. In case of doubt, don't hesitate to discuss about this with the client. (We'll probably set up a Slack channel to facilitate your discussions with the client.)

4.2.2 Post Edition

The ability for a user to edit a previously created post is currently not available. (Remember that a post is the generic news item whereas a publication is the instantiation/adaptation of that news item for a particular channel.) You are asked to add to the application this functionality for editing a post. Be careful however. Whereas the editing of a post may seem like a simple functionality, it is not so trivial since it may also involve editing the various publications that are instantiations of that post. This will require certain choices to be made regarding how to propagate these changes due to an edit. You could make certain simplifying assumptions such as that the changes won't be propagated to publications that have already been published (for many channels it would be too hard or impossible to "undo" or edit a prior publication). But that still doesn't solve the problem for the unpublished publications. For example, when I change the title of a post, should this change of the title be reflected in the title of a Facebook publication? If yes, how would you decide to implement this? And what if the title of the publication has already been edited manually before the change to the title of the post was made? Do I need to discard that manual change then? Or do I need to

replace the manually changed title by the adapted post title? Or do I need a tool to help the user chose which of the two would be most appropriate? Various solutions of varying degree of complexity can be imagined. Some freedom is left to the development team but you will have to justify your choices.

4.2.3 Post Deletion

In the same vein as the edition, you are asked to provide the functionality for deleting a post and to make a well-thought and well-justified choice for propagating (or not) this deletion to the publications related to this post. When deleting a post will you delete or keep some or all of the publications related to that post? What choice did you make? Why? How did you implement this?

4.2.4 Archival

For a user or moderator, it may not always be clear if some news item may already have been published before, so he would like to be able to access an archive containing the history of his posts, as well as the published posts of others. The archive should contain a list of posts with their data, their author, and on which channels this post was published and when. Again, some liberty is given to the implementors to design this functionality in a well-justified manner.

4.2.5 Non-validation of publication by a moderator

Currently, the application allows a user to modify the data of a publication until it is submitted for publication. The moderator functionality is not fully implemented yet. In this module you are asked to complete this functionality to allow a moderator to accept or reject a publication before it is submitted for final publication. In case of rejection the non-compliant publication is sent back to its author for further editing (or deletion). Optionally, the moderator could provide some information to the author as to why his publication was not accepted. Also, when the post is resubmitted by the user the moderator should have an indication that this is not a new post but a resubmitted one. And maybe the changes that were made with respect to the original submission should be highlighted, or a comment should be provided by the user who resubmitted it.

Think carefully about this module before implementing it, optionally discuss about it with the client, and justify your analysis and implementation choices. Again, some liberty is given to the implementors to design this functionality in a well-justified manner.

4.2.6 Alternative application features

If, in addition to the application features mentioned above, you have some inspiration for another interesting feature that we didn’t think about yet, you may also suggest your own. Feel free to discuss it with the teacher and client.

Remark

Note that the implementation of each of the above extensions may have an impact on the database structure and / or template structure of the application. Whereas you have the right to change these, any such changes should be carefully reported, documented and explained. Also, don't forget that you will need to merge your extension with that of other groups, which could lead to merge conflicts. Clearly documenting all the changes you made, especially to parts that are common with other groups, may help you to solve some of the possible merge conflicts you may encounter.

5. Choice of Modules

Each group will be required to implement the encoding of publications on (at least) 2 different communication channels (cf. the 9 channels listed in §4.2.1). In addition, they will need to implement (at least) one of the other functionality extensions mentioned in §4.2.2 until §4.2.5. Or, as an alternative to these functionality extensions, they may also provide their own (cf. §4.2.6), after prior agreement by the course teacher and the client.

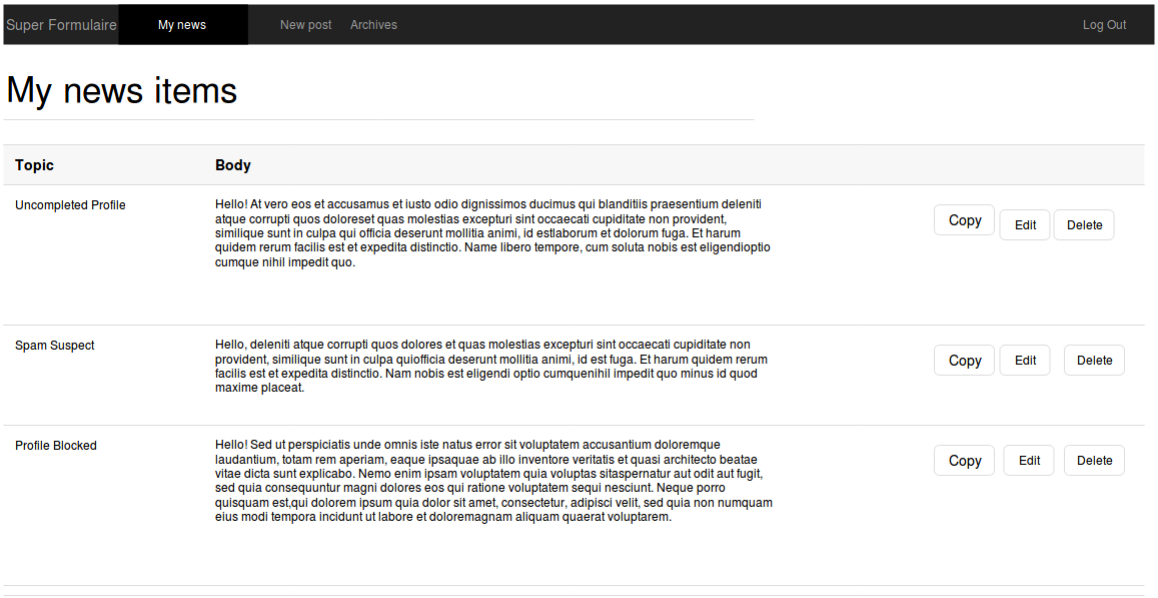
6. Submission Deadlines

Three prototypes will need to be created throughout this project. One for each of the 2 communication channels you chose, and one for the additional functionality extension. For details on these deadlines and other deadlines and requested deliverables we refer to the document explaining the project organisation that will be provided to you by the professor.

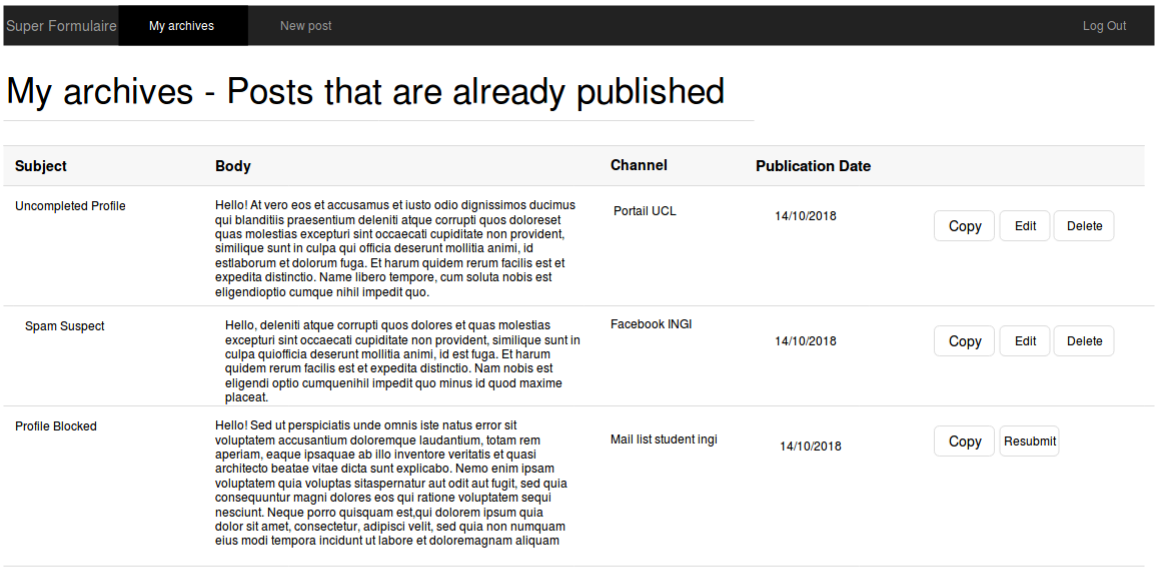
7. Appendix : User Interface Screenshots

This appendix shows some screenshots of the user interface of the different components of the current prototype of the application. A live demo of the application will be given to you during one of the first course sessions. In case things would be unclear to you, you will have the occasion to directly interact with the client at that moment. If you would have radically different ideas of interacting with the user, feel free to contact the client to discuss about this.

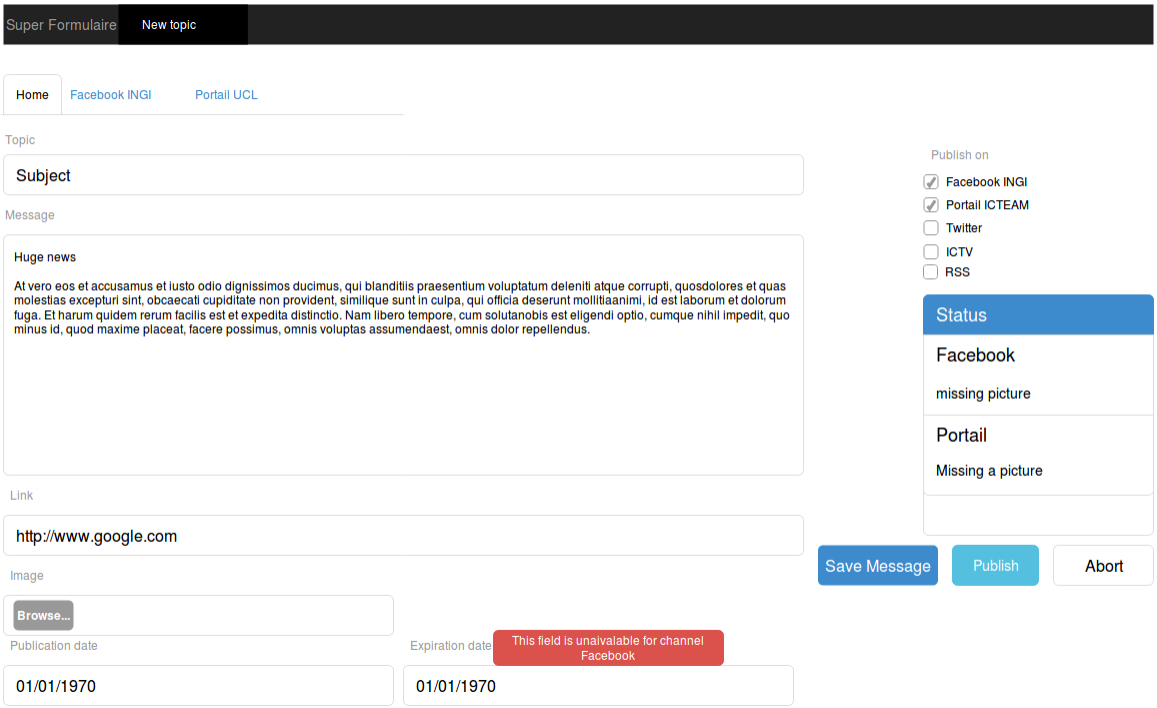
7.1 Interface – List of news



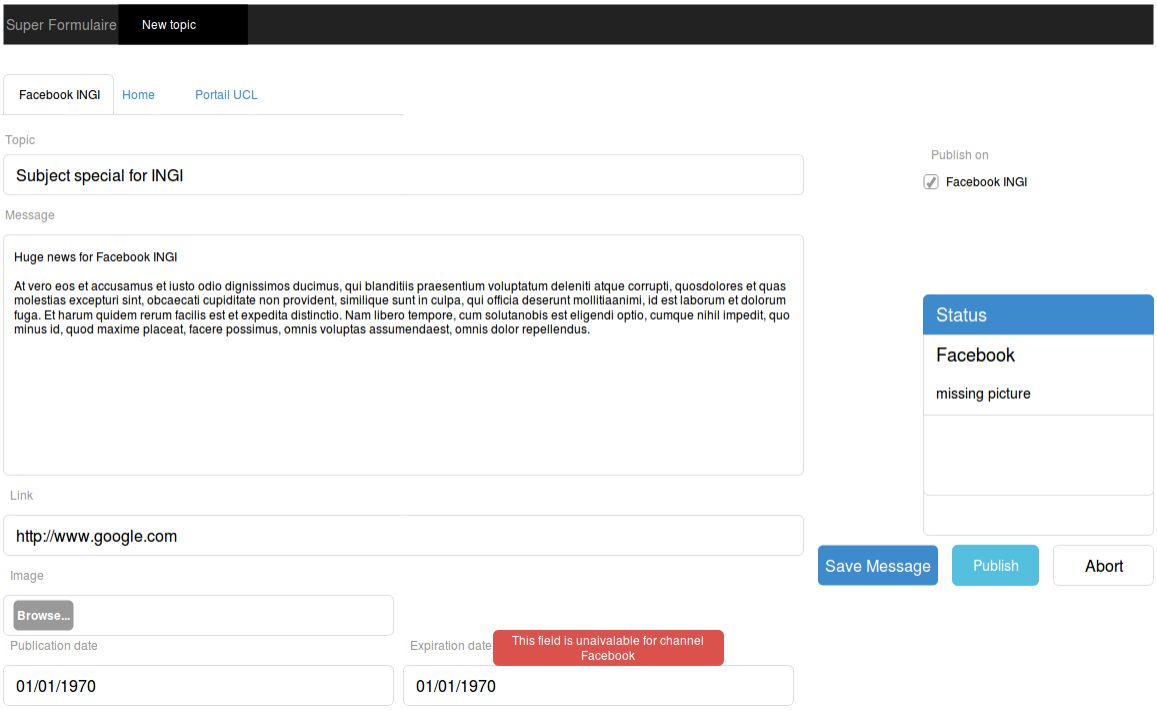
7.2 Interface – List of archives



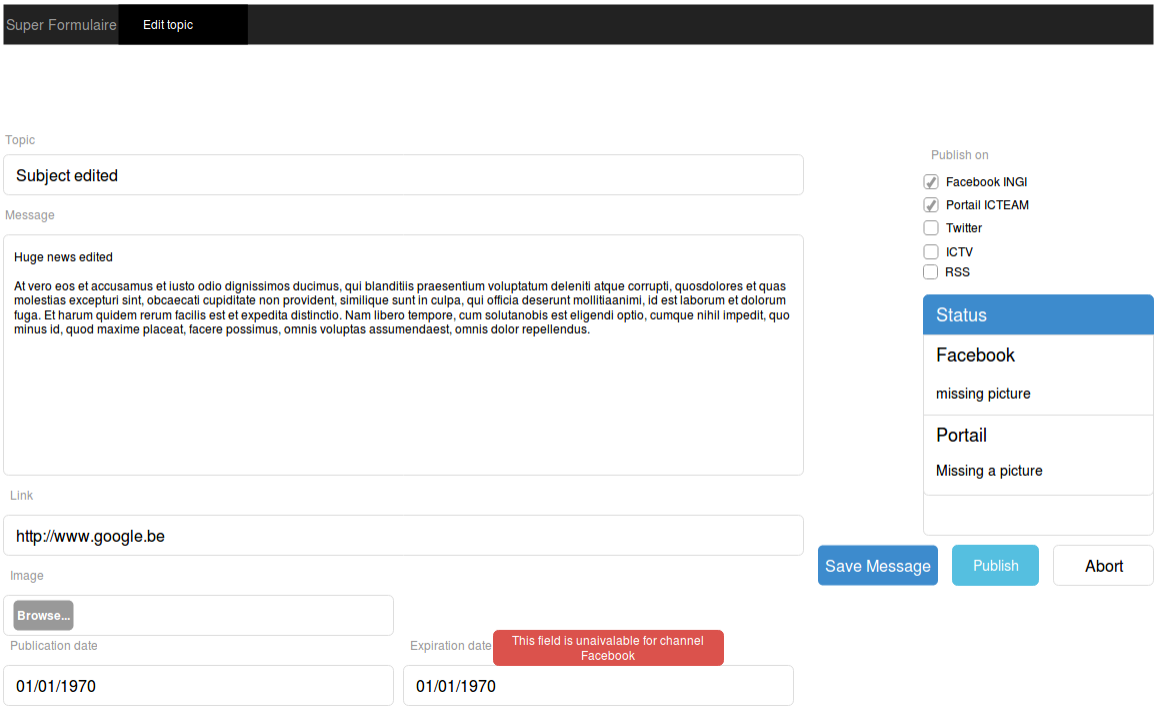
7.3 Interface – New post



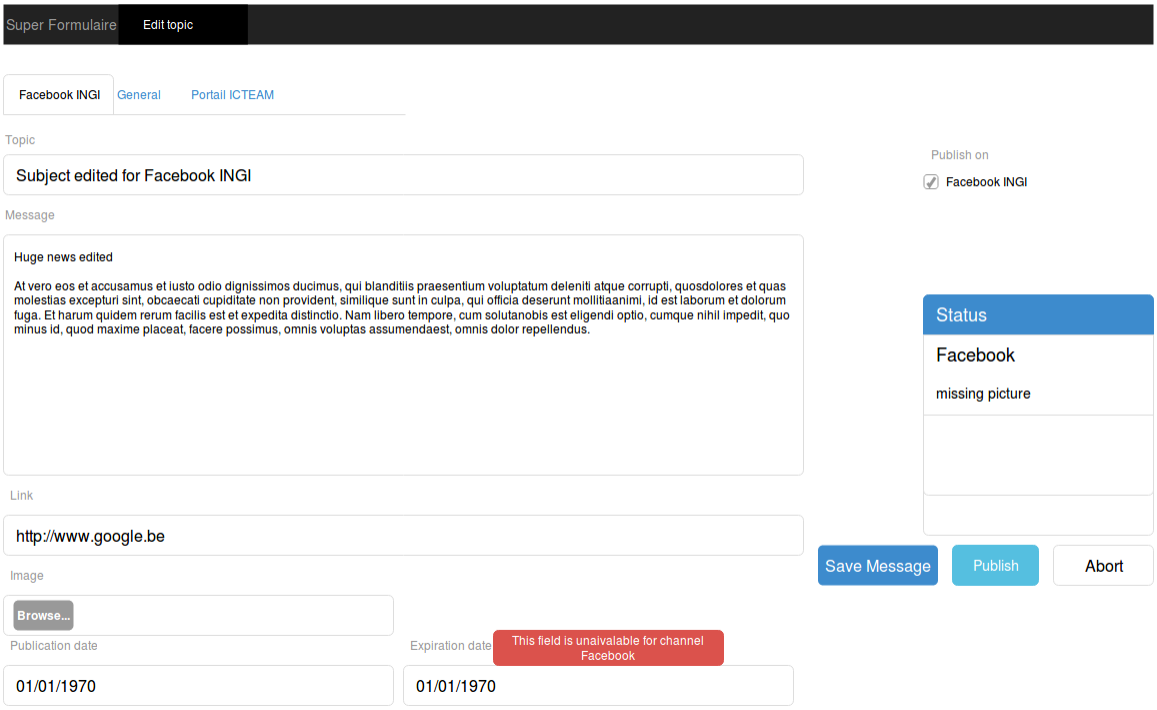
7.4 Interface – New Post : Facebook tab



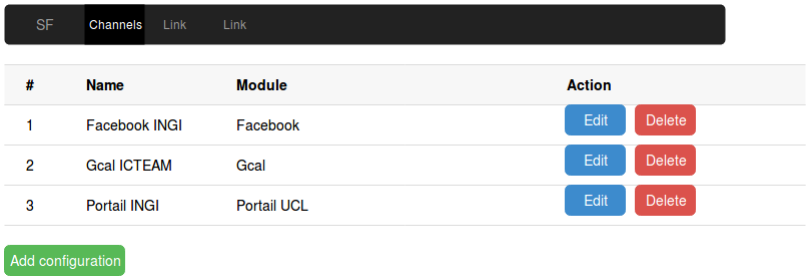
7.5 Interface – Edit Post



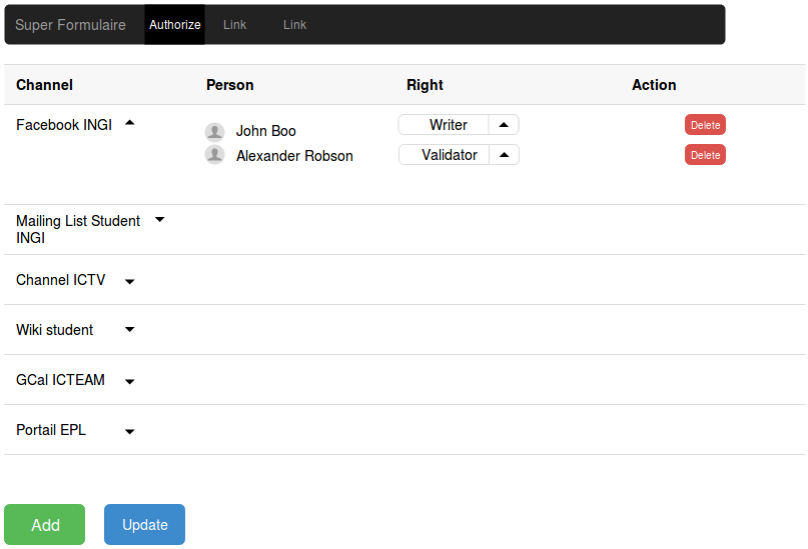
7.6 Interface – Edit Post : Facebook tab



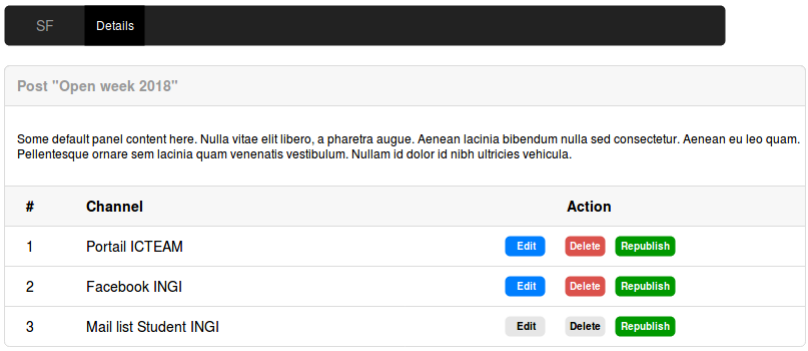
7.7 Interface – Channels



7.8 Interface – Authorize



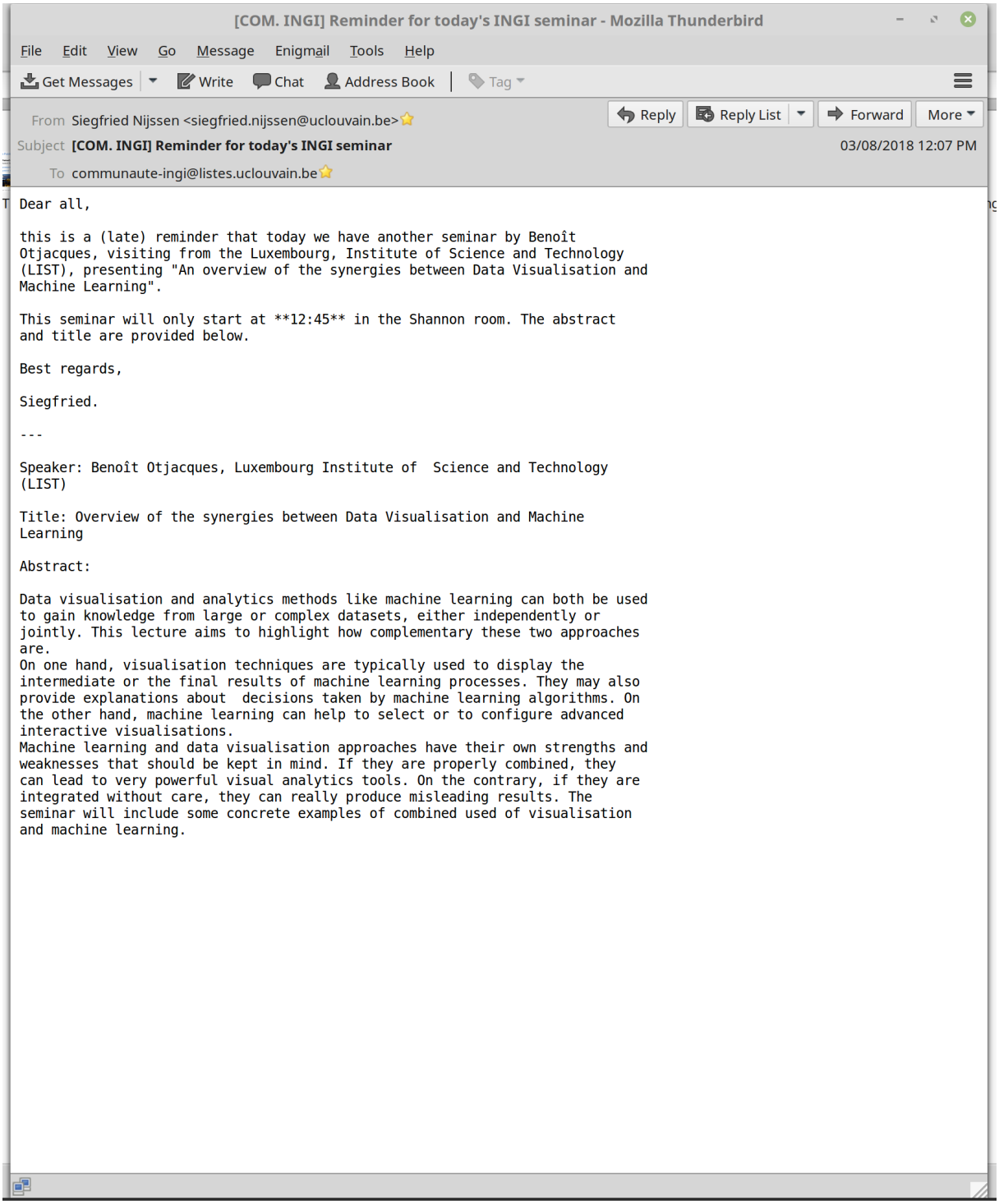
7.9 Interface – Archive details



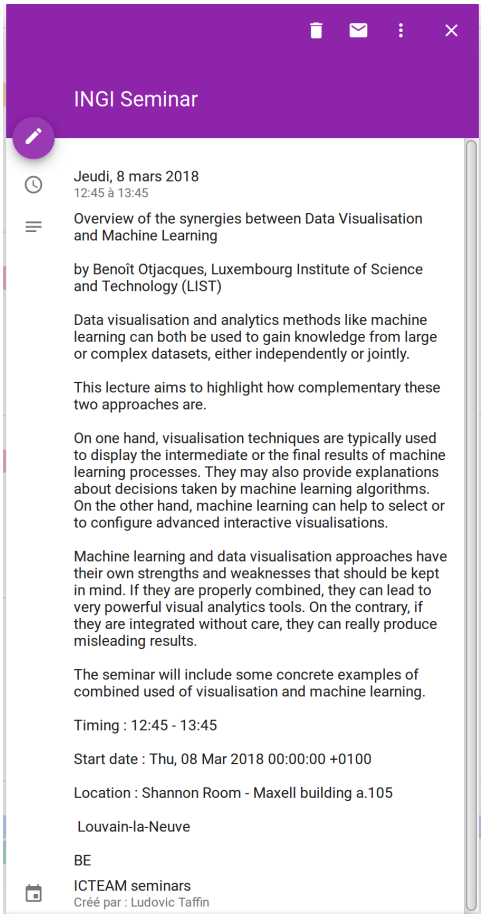
8. Appendix : Examples of possible publications and channels

To provide you with some concrete inspiration and illustrations, this appendix presents some examples of possible publications and the channels they could be posted on, taken from real life examples that have been communicated before inside our department.

8.1 Email communication about a seminar



8.2 Gcal announcement of the date of a seminar



INGI Seminar

Jeudi, 8 mars 2018
12:45 à 13:45

Overview of the synergies between Data Visualisation and Machine Learning

by Benoît Otjacques, Luxembourg Institute of Science and Technology (LIST)

Data visualisation and analytics methods like machine learning can both be used to gain knowledge from large or complex datasets, either independently or jointly.

This lecture aims to highlight how complementary these two approaches are.

On one hand, visualisation techniques are typically used to display the intermediate or the final results of machine learning processes. They may also provide explanations about decisions taken by machine learning algorithms. On the other hand, machine learning can help to select or to configure advanced interactive visualisations.

Machine learning and data visualisation approaches have their own strengths and weaknesses that should be kept in mind. If they are properly combined, they can lead to very powerful visual analytics tools. On the contrary, if they are integrated without care, they can really produce misleading results.

The seminar will include some concrete examples of combined used of visualisation and machine learning.

Timing : 12:45 - 13:45

Start date : Thu, 08 Mar 2018 00:00:00 +0100

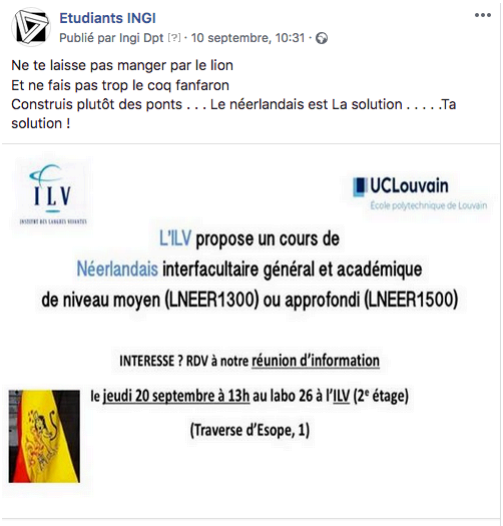
Location : Shannon Room - Maxell building a.105

Louvain-la-Neuve

BE

ICTEAM seminars
Créé par : Ludovic Taffin

8.3 Facebook post of an announcement to the students



Etudiants INGI
Publié par Ingi Dpt [?] · 10 septembre, 10:31 ·

Ne te laisse pas manger par le lion
Et ne fais pas trop le coq fanfaron
Construis plutôt des ponts . . . Le néerlandais est La solution . . . . .Ta solution !

ILV

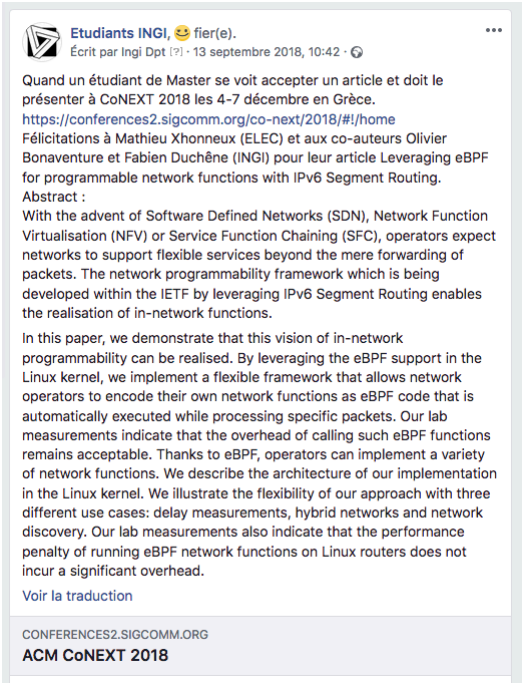
UCLouvain
École polytechnique de Louvain

L'ILV propose un cours de
Néerlandais interfacultaire général et académique
de niveau moyen (LNEER1300) ou approfondi (LNEER1500)

INTERESSE ? RDV à notre réunion d'information
le jeudi 20 septembre à 13h au labo 26 à l'ILV (2e étage)
(Traverse d'Esope, 1)

8.4 An ICTV announcement

8.5 Facebook announcement about a conference presentation



8.6 Another ICTV announcement