VRIJE UNIVERSITEIT BRUSSEL

Master thesis submitted in partial fulfilment of the requirements
for the degree of Master of Science in de Toegepaste Informatica

AUTHORING ENVIRONMENT FOR CASUAL USERS TO STEER THE PERSONALIZED DELIVERY OF LEARNING ACTIVITIES IN A MOBILE 2.0 ENVIRONMENT

PASCAL PIETERS
Academic year 2017-2018

Promoter: Prof. Dr. O. De Troyer
Advisor: Jan Maushagen
Science & Bio-Engineering Sciences

Masterproef ingediend in gedeeltelijke vervulling van de eisen voor het behalen van de graad Master of Science in de Toegepaste Informatica

AUTHORING ENVIRONMENT FOR CASUAL USERS TO STEER THE PERSONALIZED DELIVERY OF LEARNING ACTIVITIES IN A MOBILE 2.0 ENVIRONMENT

PASCAL PIETERS

Academiejaar 2017-2018

Promotor: Prof. Dr. O. De Troyer
Begeleider: Jan Maushagen
Wetenschappen & Bio-ingenieurswetenschappen

Abstract

Early school leaving (ESL) can have serious consequences. It has been shown that unemployment and poverty are two problems that are directly related to ESL. With *early school leavers*, we refer to the group of youngsters between age 18 and 24 that have left school only with a diploma of lower secondary education (or lower). Statistics of ESL indicate that ESL in the Brussels-Capital region is considerably higher than in the rest of Belgium and Europe. To address ESL in the Brussels-Capital region, a project named *Adaptive Persuasive ICT Tools to Tackle School Burnout among Youngsters in Brussels* (TICKLE) has been set up, because school burnout often precedes ESL. TICKLE aims to develop innovative ICT tools to support teachers and social workers to (re)activate youngsters that are suffering from school burnout.

This thesis is part of the TICKLE project. Our work aims to develop an authoring environment for the teachers, school supervisors, coaches and social workers that will use the TICKLE system to reactivate their youngsters. These users can be considered casual users, as these users do not necessarily have a good computer science background. The authoring environment will be used to add all relevant background information on a student, and to create short, attractive interactive learning activities that can be completed in a mobile web 2.0 environment, for instance on a smartphone. The tool will also allow to create challenges, i.e. schedule learning activities for a youngster. The tool will support the user by facilitating the discovery of learning activities that are most suitable for a youngster, in this way contributing to the personalized delivery of the learning activities. Finally, in the authoring tool it will be possible to lookup if the youngster has or has not completed the challenges assigned to him.

In this thesis we present the research and development work done for this authoring tool. We have started with research on existing tools for creating learning activities and on available technologies and standards for tracking learner's results, as well as literature studies on factors to include in the student profile, and on the representation of learning object metadata. Using the results of this research, we have identified the requirements to come to the design and implementation of the authoring tool. Finally, the usability of the tool has been evaluated by means of a user study. Although the results of the evaluation showed that the usability of the tool was acceptable for most users, we used the feedback obtained by means of the evaluation to improve the tool.

Declaration of Originality

I hereby declare that this thesis was entirely my own work and that any additional sources of information have been duly cited. I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copy-righted material, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this thesis has not been submitted for a higher degree to any other University or Institution.

Acknowledgements

I would like to express my gratitude towards my promoter Prof. Dr. Olga De Troyer for her support, guidance, help and valuable feedback. Her comments, remarks and engagement have been a great help for me. I would also like to thank my advisor Jan Maushagen for numerous useful insights, help and assistance during the accomplishment of this thesis.

I would also like to thank my girlfriend Tamara and Pommeline for their incredible patience, understanding and support throughout my studies. Far too many times the studies inhibited me from spending quality time together. As promised, I will make up for that now.

Finally, I would also like to thank my parents and grandparents, who have never lost faith in me.

Contents

List of Figures

List of Tables

1

Introduction

1.1 Context and Motivation

In 2010 the European Council rolled out their Europe 2020 Strategy to set out the targets for Europe for the next ten years (EUR-Lex - 52010DC2020 - EN, 2010). One of the formulated goals in this document expressed the ambition to reduce the average rate of European early school leavers to less than 10% by 2020. The *Early School Leaver* (ESL) has been defined as a youngster in the age range of 18-24 with a diploma of lower secondary education (or lower) as the highest attained diploma and who hasn't received education for at least one month (Europea, 2013).

Early school leaving has some severe consequences as listed in (Vlieghe, 2014). The first consequence is unemployment. Only one third of the youngsters in the ESL group were employed in 2015. This number is about the same for ESL in Belgium and Europe. The second consequence of ESL, commonly related to the first consequence, is the risk of poverty. In the Brussels-Capital region of Belgium this is an even more severe problem. Youngsters between 18 and 24 in this region represent 8,8% of the total population, while they amassed 30,5% of all living wages provided in 2015 (Steunpunt tot Bestrijding van Armoede, Bestaansonzekerheid en Sociale Uitsluiting, 2016).

Statistics of early school leaving in Belgium can be compared to figures for the whole European Union and the Belgian regions. From the start of the Europe 2020 Strategy in 2010 until 2015 early school leaving in Europe has dropped from 14,3% to 10,7% (Eurostat, 2016). This implicates that the group of early school leavers in Europe in absolute numbers has shrunk with one third in this five-year period.

The European progress in this field is significantly better than in the Belgian situation. In the same five years, early school leaving in Belgium has dropped from 11,1% to 10,1%, meaning the absolute volume of ESL only reduced by one twentieth. As compared to the rest of Belgium and Europe, early school leaving in the Brussels-Capital region is relatively high. Although ESL has been considerably reduced in the 2010-2015 period, coming from 22,1% to 15,8%, the figure of almost 16% still remains troublesome (Sacco, Smits, Kavadias, Spruyt, & Andrimont, 2016).

ESL is a complex problem. The first symptom is often school burnout. Therefore, in order to tackle the problem of many youngsters leaving school (too) early in the Brussels-Capital region, an interdisciplinary research project of the Vrije Universiteit Brussel focused on the development of persuasive ICT tools for dealing with the problem of school burnouts was setup. The project, with support of the European Fund for Regional Development (EFRO), got the name *Adaptive Persuasive ICT Tools to Tackle School Burnout Among Youngsters in Brussels* ('TICKLE research project', 2014). TICKLE aims to develop innovative ICT tools to support teachers and social workers to activate or reactivate youngsters that are suffering from school burnout.

Digital media and modern technology will play an important role in TICKLE. It has been demonstrated that new media might have the effect to resort better engagement from youngsters in spontaneous learning processes (Vlieghe, 2016). They could give a real boost to the youngster's self-confidence and the intrinsic motivation with regard to school and learning.

The TICKLE team has decided to split up the project in several smaller subprojects. The first subproject aims to set up a technological infrastructure for the personalized delivery of learning activities in a mobile 2.0 environment. A second subproject, aims to deliver an authoring environment for casual users to steer the personalized delivery of learning activities in a mobile 2.0 environment. A third subproject will make a system to do the learning analytics of the learning activities.

With this thesis I aim to contribute to the second TICKLE subproject mentioned.

1.2 Research Goals

Because the nature of our research, we decided to follow the Design Science Research Methodology (Peffers, Tuunanen, Rothenberger, & Chatterjee, 2007). Design science is the act of creating an explicitly applicable solution to a problem. Wieringa (Wieringa, 2009) has formulated a number of guidelines to define the research goals in the problem investigation phase of Design Science. A clear distinction has to be made between practical problems and knowledge questions. Practical problems tend to make an actual change in the world, while knowledge questions call for a change of our knowledge of the world. Since we were asked to build a new platform, i.e. an authoring environment for casual users to steer the personalized delivery of learning activities in a mobile 2.0 environment, we can state that our main problem is a practical problem, which will indeed result in a change in the world. However, to solve this practical problem we will need to solve some knowledge questions. We will now define the main practical problem and the related knowledge questions.

Main practical problem

Build an authoring environment for casual users to steer the personalized delivery of learning activities in a mobile 2.0 environment.

As Wieringa points out, all solutions to practical problems should meet the requirements of the stakeholders. In our case the stakeholders, called *casual users* in the main practical problem, will be educators and/or social workers. Educators and social workers are considered to be casual users in the sense that they do not necessarily have a deep understanding of computer systems, which implicates that usability will be a major concern.

Knowledge questions

To come up with an acceptable solution to the formulated main practical problem some knowledge questions need to be answered first.

RQ1: Which are the existing technologies and standards that can be used for learning activities in a mobile web 2.0 environment and what are their strengths and weaknesses?

Before proposing a solution for the main practical problem, it will be important to know which technologies are available for specifying learning activities (also called learning objects) in a mobile web 2.0 environment and which are the current standards being used. Opting for a standard is important to enable interoperability with other learning systems. Once we have an overview of the current technologies and standards, we will be able to choose the most suitable solution for our practical problem.

RQ2: *What are the existing authoring tools that can be used to create learning activities in a mobile web 2.0 environment and what are their strengths and weaknesses?*

A whole plethora of authoring tools for creating mobile learning activities already exists. Since we don't intend to reinvent the wheel, we will have to investigate the existing solutions for their use in our tool.

RQ3: Which information should the student profile contain?

The main practical problem states that delivery of learning activities should be personalized. The standard way to achieve personalization is by using user profiles that describe for each user its preferences and characteristics. In our system, the users are the students and therefore the system will have to contain a student profile of all its students. Research should be done on what information should be part of this student profile in relation to the main practical problem we are trying to solve.

1.3 Research Methodology

The research methodology used to come to the solution for the main problem and sub-questions is the Design Science Research Methodology (DSRM) for Information Systems Research (Peffers et al., 2007). Design science is the act of creating an explicitly applicable solution to a problem. The DSRM incorporates principles, practices, and procedures for design science. The methodology consists of six activities leading up to the requested solution: (1) problem identification and motivation, (2) objectives for a solution, (3) design and development, (4) demonstration, (5) evaluation and (6) communication. We will briefly describe how we have dealt with each of these six phases of our design research.

1.3.1 Problem Identification and Motivation

The problem identification and motivation have already been addressed in sections 1.1 (*Context and Motivation*) and 1.2 (*Research Goals*). It has been made clear that ESL is troublesome in the Brussels-Capital region and increases the risk of unemployment and poverty. School burnout often precedes the actual school leave. Therefore, we will be tackling school burnout by developing a platform that allows sending appealing learning activities to possible dropouts. As describe above, this thesis will only address the authoring environment, thus contributing to the main goal of TICKLE. The problem identification is made concrete in the formulation of the main practical problem in section 1.2.

1.3.2 Objectives for a Solution

Finding the answers to research questions RQ1, RQ2 and RQ3 will allow us to propose a solution to the main practical problem. Before we can come up with a solution, we need to have a clear view on existing authoring tools and well-known and accepted standards and technologies for learning activities. We will also have to identify the structure of the student profile in order to come up with an acceptable system for the problem stated. A report on the research done and answers formulated for the research questions RQ1, RQ2 and RQ3 can be found in chapters 2, 3, 4 and 5. The answers to these knowledge questions have helped to define the objectives for the solution (i.e. the system requirements), which are given in chapter 6.

1.3.3 Design and Development

Using the formulated requirements, the design and implementation of the system could start. Since the contribution of this thesis is only a part of the complete authoring tool to be developed, it was necessary to first determine how our part of the authoring tool will be connected to the rest of the TICKLE system. We then made choices for the technologies we would use for our part of the solution. We chose for the PHP CodeIgniter 3 web application framework. For the software architecture, we opted for a Model-View-Controller (MVC) approach. The proposed design and implementation are discussed in chapter 7.

1.3.4 Demonstration

The solution has been demonstrated with a number of examples. Live demonstration has been limited to a session with members of the TICKLE project. There were also a number of videos recorded to demonstrate a general walkthrough of the workflow of the proposed solution. These videos have been made part of the user manual, which can be accessed within the application. Together, these videos provide an overview of all functionality that is available in our solution.

1.3.5 Evaluation

An evaluation is done in order to determine whether the requirements are met. In chapter 6 different kinds of requirements for the solution have been formulated: a list of functional requirements, a short list of non-functional requirements (non-usability) and an elongated list of usability requirements. Since the application is intended to be used by casual users, the focus in the evaluation has been on the usability requirements. For this purpose, a usability evaluation with users from the target audience has been done.

In formulating the usability requirements we have already included the measuring concept and acceptable levels. These formed the basis for the evaluation. The evaluation was carried out with ten participants, all teachers in a secondary school. For the evaluation we created eight user tasks. Together these eight user tasks are representative for the main functionality of the tool. Cognitive walkthrough in combination with think aloud was used for collecting data when the participants execute the user tasks. Afterwards a short post-session discussion was done and the participants were asked to fill out a questionnaire. This provided enough data to evaluate the usability requirements that were defined. A report of the evaluation and its results can be found in chapter 8.

1.3.6 Communication

This thesis report can be considered as the document that reports all findings of our research. It will be made freely available via the WISE website.

1.4 Thesis Structure

In this first chapter we started by explaining the context and motivation of the thesis subject. We have continued by setting out the research goals and explaining what methodology was used to attain our goal.

In chapter 2 we will examine the existing technologies and standards that can be used for learning activities. We will discuss the main differences between the two main standards, xAPI and SCORM. This will permit us to find out which one will be the better choice for our purpose.

Next, in chapter 3, we will look at related work, i.e. existing authoring tools. We will examine their strengths and weaknesses to determine which ones might be useful for our purpose.

In chapter 4 we deal with RQ3, i.e. the definition of our student profile. For this purpose, we will report on existing work related to the definition of user/learner profiles. Research done in the field of adaptive learning and m-learning will proof to be useful for this. Based on this we will determine what should be included in our student profile.

The next chapter, chapter 5, deals with the representation of the metadata for our learning objects. This metadata is necessary to allow for searching and reusing learning objects. By means of a literature study, we examined existing knowledge on the topic of learning object metadata. The outcome of this research results in determining the standard we will use for representing the metadata of our learning objects.

In chapter 6 we will define the requirements for our application. This will result in a list of functional requirements, non-functional requirements and usability requirements.

In chapter 7 we will take up the design and implementation of the prototype.

Chapter 8 discusses the evaluation of the prototype. As stated earlier, the focus here has been on the usability of the system.

The final chapter of this document, chapter 9, contains the conclusions of our work and a discussion on possible future work.

2

Existing Technology and Standards For Learning Activities

In TICKLE, the results of the learning activities that are presented to the students need to be recorded. Therefore choices will have to be made on how and where to store these results. In this chapter we will elaborate on the two main current standards and technologies for tracking and tracing the results of learning activities: SCORM and xAPI. After discussing the features and limitations of each standard, we will make a comparison. Subsequently we will explain why we are convinced that using xAPI is the better option for our needs.

Finally, we will give an overview on the different available systems and architectures to store learning activity results with the xAPI standard. After this overview, we will present and explain our choice of storage.

2.1 SCORM: Shareable Content Object Reference Model

SCORM (Shareable Content Object Reference Model) has been the de facto standard for the creation of e-learning software since the year 2000 ('Sharable Content Object Reference Model', 2017). ADL (Advanced Distributed Learning), a research group sponsored by the United States Department of Defence, composed the SCORM standard, aiming to enhance the sharing of reusable learning objects.

The basic idea with SCORM is that a teacher can create a learning object that can be used in different learning management systems. All learning management systems (LMS) on the market nowadays make use of SCORM learning objects. These can be created by the teachers and inserted into the system. When students use the learning objects to study and take quizzes, their scores will be recorded in the LMS and can be accessed by the teachers.

The SCORM standard has been developed with three main goals in mind: interoperability, portability and reusability (ADLNET, 2017a). Interoperability has been established by providing an API to learning management systems to handle communication with SCORM objects. Giving users the option to export their SCORM learning objects as zip files results in the creation of portable objects. Since all learning management systems adopted the SCORM standard reusability of the objects was achieved.

The underlying technologies that have been used to facilitate interoperability, portability and reusability are XML and JavaScript. All learning objects are .zip-files, containing XML-files that describe the SCORM widget functionality. The complete description of the XML-schema used in the different versions of SCORM is freely available, see (Rustici Software, 2018h). JavaScript is used to generate interactivity in the browser.

The SCORM standard has several versions, starting with SCORM 1.0 in the year 2000. SCORM 1.2, released in 2001 is the first version of SCORM that was widely adopted. In 2004, SCORM began to release different editions of SCORM 2004 based on iterative fixes and improvements. The most recent release (2009) is SCORM 2004 4th Edition.

2.2 xAPI/Tin Can

2.2.1 The need for a new standard

A lot has changed since the introduction of the SCORM standard. E-learning is no longer limited to learning objects in a browser launched from within an LMS. No, learning anno 2018 is happening everywhere: on mobile phones, games, real world experiences, collaborative learning …

Since learning has evolved that much, around 2010, it was time to re-evaluate the standard for e-learning and replace it with a standard being up-to-date with the latest view on m-learning. In 2010, ADL started the search for a successor of SCORM by gathering input from all partners in the field of learning (Rustici Software, 2018b).

SCORM had its merits and those would be kept in the new project. The new standard, just like SCORM, should provide functionality to track some 'basics' of learning experiences: completion, time, pass/fail and score.

The goal was to come up with a new standard that should make up for some of the drawbacks that were experienced with SCORM in an ever evolving world of learning (Rustici Software, 2018e):

- SCORM learning objects can only be launched from within a LMS
- SCORM reduces learning to following some pre-determined steps that lead to a score
- SCORM objects always require an Internet browser to be executed
- SCORM learning experiences are limited to one domain (LMS) and can only be accessed within that system

The new Tin Can standard addresses these SCORM limitations by:

- Enabling to launch learning experiences outside an LMS
- Enabling whole new ways of learning (for instance new video of Khan Academy), not limited to following predefined paths
- Enabling learning outside an Internet browser, for instance in custom applications
- Enabling cross-domain access to the stored learning experiences, not limited to one LMS

On top of that the new standard aimed to extend the concept of learning as broad as possible:

- Use of mobile apps for learning
- Track informal learning and real-world experience
- Track off-line learning (local storage until connection established)
- Track team-based and interactive learning
- Track serious games and simulations

2.2.2 The basic idea of xAPI/Tin Can? (Rustici Software, 2018c)

The basic underlying idea of the new standard is quite simple: people learn by interacting with text, video, e-content, other people … The aim of the new standard is to provide a means of recording learning interactions and store them in a Learning Record Store (LRS). Every interaction is stored by sending a secure statement to the LRS in the form of "Noun, verb, object" ("I did this"). An example of such a statement could be: "Pascal finished exercise 5". All of these data can be accessed, within or outside an LMS. LRSs can also share their data between each other.

This standard has a whole new philosophy about learning. Learning is no longer limited to working your way through a pre-made learning object, but literally 'everything' can be recorded as a 'learning activity': watching a video, attending a conference, following a step-by-step tutorial on the internet, reading a book, writing a paper... A learner is no longer bound to the limits of a browser and an LMS to register his learning activities. Even better, he doesn't even have to be online to have it recorded (it can be done afterwards). All data are recorded in the form of short sentences.

The original version of the new standard was designed by Rustici Software and was called Tin Can Project. With the official release later, ADL changed the official name to xAPI (Experienced API). In learning environments however, the name Tin Can had already been well established and many people kept using it. Because of that, both names (Tin Can and xAPI) are now used for the same thing. There is no difference between Tin Can and xAPI.

2.2.3 **xAPI/Tin can statements (Rustici Software, 2018k)**

The xAPI protocol is used to send statements to the LRS to store them. Afterwards they can be retrieved to perform some analytics. xAPI defines how statements should look like to be accepted by the LRS. In the simplest form, a xAPI statement is of the form 'Actor Verb Object', e.g. 'Pascal has read "The catcher in the rye"'.

All statements of this kind should be sent to the LRS in JSON-format, i.e. a number of properties expressed as key/value pairs. An example statement could look like this (Rustici Software, 2018j) :

```
{
        "actor": {
            "name": "Example Learner",
            "mbox": "mailto:learner@example.com",
            "objectType": "Agent",
        },
        "verb": {
            "id": "http://adlnet.gov/expapi/verbs/completed",
            "display": {
                "en-GB": "completed",
                "en-US": "completed"
            }
        },
        "object": {
            "id":     "http://www.example.com/some-checklist-id/some-checklist-item-
id",
            "definition": {
                "name": {
                    "en-GB": "example to-do list item",
                    "en-US": "example checklist item"
                },
                "description": {
                    "en-GB": "A more detailed examplation of the item completed.",
                    "en-US": "A more detailed examplation of the item completed."
                },
```

```
                "type": "http://id.tincanapi.com/activitytype/checklist-item",
            },
            "objectType": "Activity"
        },
        "result": {
            "success": true,
            "completion": true,
            "duration": "PT1H37M56.34S"
        },
        "context": {
            "registration": "ec531277-b57b-4c15-8d91-d292c5b2b8f7",
            "contextActivities": {
                "parent": [
                    {
                        "id": "http://www.example.com/some-checklist-id/",
                        "objectType": "Activity",
                        "definition": {
                            "type":
"http://id.tincanapi.com/activitytype/checklist"
                        }
                    }
                ]
            },
            "language" : "en"
        },
        "stored": "2013-05-18T05:32:34.804Z",
        "authority": {
            "account": {
                "homePage": "http://cloud.scorm.com/",
                "name": "anonymous"
            },
            "objectType": "Agent"
        },
        "version": "1.0.0",
        "id": "6690e6c9-3ef0-4ed3-8b37-7f3964730bee",
        "timestamp": "2013-05-18T05:32:34.804Z"
    }
```


The minimal statement to send to the LRS should have at least the main properties actor, verb and object. On top of that, it can (but must not) also include the properties result, context, attachments and/or statement information (stored, authority, version, id, timestamp).

Each of these main properties in the JSON-object can also have a number of predefined subproperties. We will limit ourselves here to pointing to the example above to get a picture of the subproperties that will be mostly used. For a complete description of the statements, we refer to the Tincanapi website[1].

For every subproperty 'id', the associated value has to be an URI. Since it is not a good idea for everyone using different URI's with the same meaning, a standard vocabulary for these URI's has been agreed upon. The Tin can API registry has been developed to provide standard URI-values for the most common key properties. This registry has been made available to the public by the community (ADL, 2017).

[1] https://experienceapi.com/

Standard URIs are available for the following property ids of a Tincan statement: Activity types, Attachments, Extensions, Verbs and Profiles. For instance, in the example above, the URI used for the verb 'completed' is **http://activitystrea.ms/schema/1.0/complete**. This is the URI from the Tincanapi registry for this verb.

2.2.4 Accessing xAPI data in the LRS (Rustici Software, 2018f)

The added value of xAPI over SCORM is not only to be found in the recording of data. Evenly important is the provided access to the data. Unlike SCORM, the xAPI standard provides rules for an LRS on how to make its data accessible.

In the old SCORM standard, data could only be accessed through the LMS and users had to be satisfied with the way the LMS presented the data to them. With xAPI being a standard not only for writing to a LRS, but also for reading data from it, the data become freely available to the user. This means that custom applications can be designed to present the data to the user in the desired format. This could take reporting and learning analytics to a whole new level.

Unlike with SCORM, data can be written to more than one database. It is perfectly possible to write data of your learning experiences to the LRS of your teacher/boss and to your *personal data locker*, your personal LRS, keeping track of all your learning experiences. This personal data locker can be accessed at all times to reflect on the own learning or for solicitation purposes.

In fact, the LRS enables us to record all of a persons' Activity Streams (Rustici Software, 2018g). Just like, for instance, Facebook records all their users activities (e.g. 'Tom liked your post'), an LRS can record all learning, training or other work-related experiences of their employees or students. Some early adopters are already using this approach to record experiences of employees and compare the outcomes of different trainings. This can help them in their quest for the best training for the job. In the same way, teachers can use an LRS for their students' Activity Streams and compare the results. It can help them determine the optimal learning path.

2.2.5 xAPI/Tin Can system architecture (ADLNET, 2017b)

xAPI is setup as a Restful web service, passing JSON formatted data from and to the LRS. xAPI provides a standard for all communication (reading and writing) with the LRS. Rustici and ADL have provided 4 different APIs for this. These are software libraries that programmers can use to handle all communication with the LRS in the form of Restful http methods. The software libraries have been made available for 8 platforms and programming environments (Rustici Software, 2018a).

2.2.6 Versions of xAPI or Tin Can (ADLNet, 2017)

- 2012: versions 0.9 and 0.95
- 2013: version 1.0 and 1.0.1.
- 2014: version 1.0.2.
- 2015: version 1.0.3.

2.3 Comparison Table: SCORM vs. xAPI (Rustici Software, 2018i) (iSpring Solutions, 2014)

e-Learning options	SCORM	xAPI/TinCan
Tracks completion	Yes	Yes
Tracks time taken to accomplish a task	Yes	Yes
Tracks pass/fail (complete/incomplete)	Yes	Yes
Reports a final score	Yes	Yes
Reports multiple scores (multiple attempts)	-	Yes
Detailed test results (e.g., answer breakdown)	Partially [1]	Yes
Solid security	-	Partially [4]
Works outside an LMS	-	Yes
Works without an Internet connection	-	Partially [5]
Keeps complete control of your content	-	Yes
Free of cross-domain limitations	-	Yes
Allows using mobile apps for learning	Rarely [2]	Yes
Platform transition (i.e. from computer to mobile)	Partially [3]	Yes
Tracks simulations	-	Yes
Tracks real-time performance	-	Yes
Tracks offline learning	-	Yes
Tracks long-term learning	-	Yes

Table 1 Comparison Table Scorm vs. xAPI. Reprinted from iSpring Solutions (2014).

1 – Many LMSs do not parse cmi.interactions.n.id for SCORM 2004 quizzes, and SCORM 1.2 is not designed for essay questions.

2 – Most LMSs do not have mobile apps that are compatible with SCORM.

3 – Many authoring tools do not have good HTML5 support to create SCORM courses for mobile devices.

4 – It depends on where the content is located.

5 – To establish a connection with an LRS, an Internet connection is required if xAPI courses are launched in a browser and not in a specially designed application.

2.4 The Learning Record Store (LRS)

Basically an LRS is a system that deals with storing of learning information and learning information retrieval. All learning activities are stored in the LRS by sending statements to it (illustrated in Figure 1). The saved statements can later be retrieved from the LRS for analytics purposes.

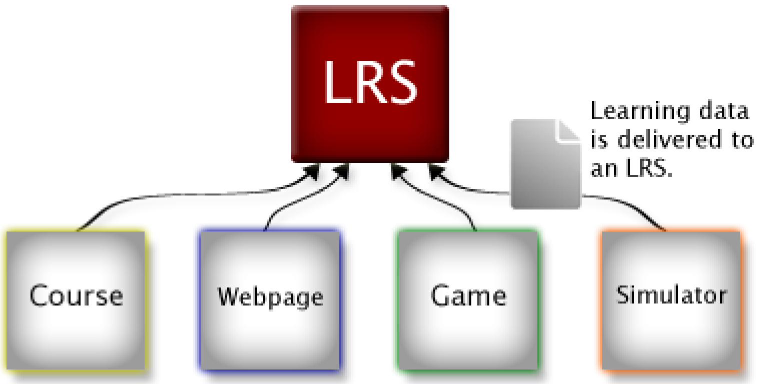


Figure 1: LRS as Data Store. Reprinted from Learning Record Store: What is an LRS? (2011).

The LRS can, but must not, share its stored data with external tools, for instance other LRSs, one or more LMSs, reporting tools (illustrated in Figure 2).

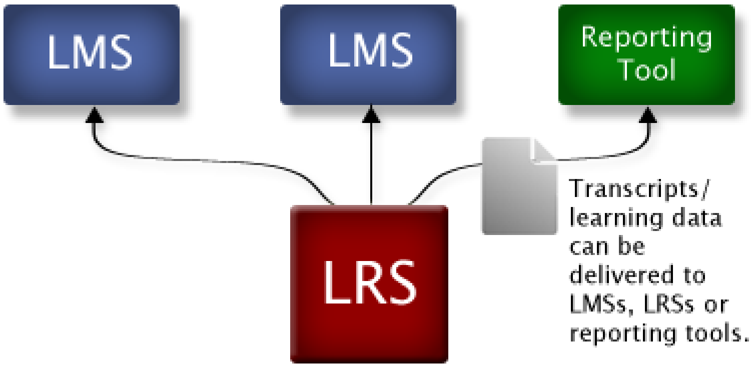


Figure 2: LRS for Sharing Data. Reprinted from Learning Record Store: What is an LRS? (2011).

2.4.1 Categories of LRS Systems

We distinguish four categories of LRS systems (Berking, 2015) . We will discuss them in this section.

- ***LRSs without data analytics engines***
  These types of LRSs just provide the core functions one might expect in an LRS, i.e. storing and retrieving data in the form of xAPI statements. They have no functionality to do some data (or learning) analytics. That functionality is supposed to be provided by the custom system using the LRS. Examples of this kind of LRS are ADL (ADLNET, 2018) and WaxLRS ('Wax LRS', 2013). The first is an open-source solution, developed for testing purposes. The latter is a commercial system that also provides an external data analytics engine as external plug-in.

- ***LRSs with integrated data analytics engines***
  A number of more enhanced LRSs provide functionality that goes beyond the mere storing and retrieving of data. They provide means to combine or aggregate, manipulate and visualize the data in the LRS. They also let users tailor the data analytics to their personal requirements. A lot of commercial products are available on the market in this segment, the most popular being GrassBlade ('GrassBlade', n.d.) and WatershedLRS ('Watershed', n.d.). Next to all these commercial LRS systems, also one open-source product is available: Learninglocker ('Learning Locker', 2018). This is a solution to host the LRS on your own server.

- ***LMS with integrated LRS capability***
  A lot of LMS providers have been making the transition to using a LRS recently. Most of them have chosen the option to build an LRS into the LMS they provide (see figure 3 for an illustration). Some have developed their own LRS, while others have used 3rd party providers like those mentioned above. We will only mention here these big names: Adobe Captivate Prime ('Adobe Captivate', 2018) and Elements ('Riptide Elements', n.d.).

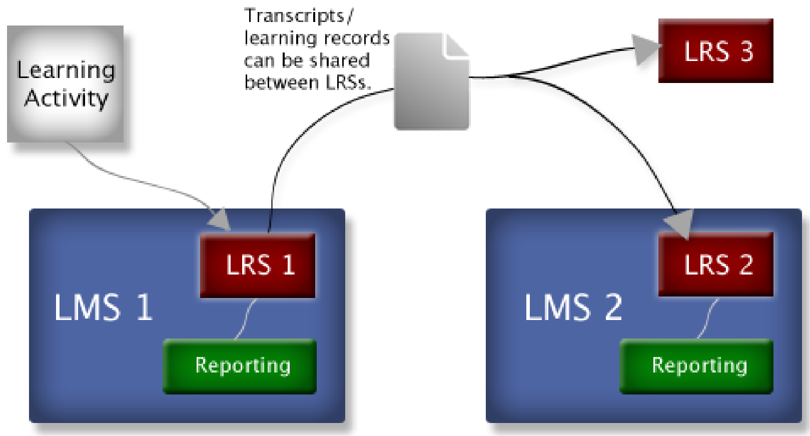


Figure 3: LMS with Integrated LRS. Reprinted from Learning Record Store: What is an LRS? (2011).

- ***LMS with API-based integration with an external LRS***
  We have stumbled upon one commercial LMS that provides API-based access to an external LRS. In this model the user can use an LRS of his own choice and configure the LMS to store certain statements in this external LRS. All communication between LMS and LRS is handled through API's. The vendor of this type of product is LearnUpon ('LMS: Learning Management System Online Training Software', 2018).

2.4.2 Choosing our LRS: Free vs Open-source, Self-hosted or Not

When choosing an LRS for the system, the first choice one has to make has to do with the hosting. One can choose to go with a cloud based solution where the provider offers a fully fledged hosted solution to the users. The alternative is to go for a complete self-hosted LRS on your own server.

As mentioned above, there are of course a lot of commercial providers offering cloud based solutions, but a number of free alternatives can also be found ('Learning Record Stores', n.d.). The most well-know free cloud based LRS is SCORM Cloud (Rustici Software, 2018d). This service, offered by Rustici, is free for up to 10 registrations and 100MB of data. There is also the before mentioned ADL LRS, but that one can only be used for testing purposes. It is a great LRS to quickly start trying things out and doing some basic first testing by sending statements to it. Learninglockers cloud LRS is free to try out for 14 days.

For anyone looking for full control, a self-hosted LRS would be the way to go. ADL is of course free, but cannot be used in a production environment. Sakai has also released a Java-based LRS under the ECL2.0 license. At the time of writing (April 2018), the code still had some issues (Apereo, 2016). The only remaining solution to host your own LRS is Learninglocker. You can download the code and install the Learninglocker LRS on your own server. It was released under GPL3 as open-source software.

The TICKLE project will include a learning analytics module (outside the scope of this thesis). When developing the learning analytics module, Learninglocker might be considered to use, since this open-source software provides some built-in learning analytics solutions. Since 2014, the JISC project has been started to develop an open learning analytics framework for Learninglocker (HT2, 2018).

For this master thesis we preferred to use a free LRS. Most LRSs available are commercial products, but we already mentioned two free solutions: ADL and Learninglocker. For our purpose the ADL LRS will be sufficient. It can be used as a full testing environment, having all features you can expect from an LRS.

2.5 Conclusion

In this chapter we have compared the two main standards for the creation and tracking of e-learning objects: SCORM and xAPI. SCORM aims for the portability and reusability of learning activities. It has been the de facto standard since the year 2000. The latest version, SCORM 2004 4th edition, has been released in 2009.

The xAPI/Tin Can standard has been introduced in 2010 to address some shortcomings in SCORM. With xAPI learning objects, one is no longer tied to a Learning Management System (LMS) to record outcomes of learning activities. Furthermore, with xAPI learning can take place outside the browser, for instance in mobile apps. In a comparison table, we have demonstrated that xAPI overcomes all SCORM limitations and adds a whole range of new features for tracking and reporting outside an LMS.

We have also reported on the system architecture of xAPI: providing web services to read and write the Tin Can statements as JSON objects to a Learning Record Store (LRS). These statements will need to follow the Tin Can controlled vocabulary.

Since we will be working in mobile web 2.0 environment and because the overwhelming amount of features xAPI has, we have chosen to use xAPI over SCORM. We strongly believe xAPI will be the standard of the future for tracking learning activities.

Finally, we have investigated the different types of LRS systems. We have demonstrated that the use of ADL's solution as LRS will be sufficient for our needs. Furthermore, using the ADL LRS won't provide any limitations if the need would arise to switch to another LRS.

3

Authoring Tools

In the context of e-learning, the term authoring tool refers to a program that enables users, usually teachers, to create e-courses (or learning objects) without the need for any real programming skills. Therefore authoring tools usually consist of drag-and-drop functionality and pre-defined templates to be filled out to create course content. In general, the created learning object may contain text, graphics, video, interactive exercises...

The three main components of an authoring system are: content creation and organization, control of content delivery, and type(s) of assessment ('Authoring system', 2017). In this section we will focus only on the content creation capabilities of existing authoring tools. As stated earlier, for our project we will create our own organization, delivery and learning analytics modules. We will however not try to reinvent the wheel and will examine if we can use existing systems for content creation. In order to do so, we first need a good overview on what's available. That is the scope of this section.

A great number of programs are available for creating your own learning objects. Since it is not feasible to discuss all of them, we have limited ourselves to some of the more popular tools used in the field of education and in a business context. For companies, authoring tools are a great opportunity to provide time and location independent online training to their employees.

We will start by discussing the common features and the main differences of those tools. Next we briefly discuss the tools considered, grouped by their availability: free open-source tools and commercial tools.

3.1 Common Features and Differences between Authoring Tools

All tools for creating learning objects we have reviewed have been created for educators. In this section, the term *educator* not only refers to teachers, but also to company employees that create courses for online in-company training. Since these users normally do not have a computer science background, the creation of learning objects should not require any programming skills. Therefore it usually happens through a graphical user interface (GUI). In such a GUI the learning object can be created by combining video, audio, images and text into a single learning object.

The educator can create a learning object by using the building blocks provided by the tool. All presented authoring tools provide functionality to present content to the end-user in many different formats: text, pictures, video, charts... Most tools in this paragraph also allow creating interactive learning experiences. The ones they all have in common are: quiz assessment, multiple-choice questions (single answer or multiple answers), gap fill exercises and matching exercises.

Differences can be found in the additional functionality an authoring tool provides. Besides the common ones already mentioned, some of the tools have more possibilities for interactivity in learning objects: timeline/ordering, drag-and-drop labelling or classical games like Hangman or the classical Memory Game. One critical difference in the tools available is the export possibilities. Most tools provide the possibility to export the created learning object in SCORM 1.2 format and SCORM 2004 Flash format. Some more recent tools also provide HTML5 export, which will be the technology needed in our project. In the short review below, we will focus mainly on the differences between different existing software solutions for creating learning objects.

3.2 Open-source (Free) Authoring Tools

- **Xerte** (‘Xerte’, 2018)

Xerte is a free open-source solution that provides teachers and educators with the possibility to create a wide range of learning objects without the need of any HTML or JavaScript knowledge. The Xerte project was started by the University of Nottingham in 2004. Since 2009, a PHP/MySQL web-based self-hosted solution was released under the name ‘Xerte Online Toolkits’. In 2014, the project was handed over to the Apereo Foundation (‘Apereo’, 2018) for future development. Xerte allows users to create learning objects that contain different kinds of interactivity, e.g. Hangman and Memory Game objects. With Xerte Online Toolkits teachers can be granted access to the common environment of the authoring tool. In that environment they are also enabled to share their created learning objects with others and use those created by colleagues. Xerte supports both HTML5 and Flash export (latest version 3.5) for its SCORM objects. At the time of writing, there is no support for xAPI tracking of student results.

- **H5P** (‘H5P’, n.d.)

H5P is a community driven project, based on the enthusiasm of some volunteers and with a number of companies doing the heavy lifting. It started out in 2012 by the Norwegian company *Joubel* and has been growing ever since. H5P is a completely free and open technology, licensed with the MIT license.
Similar to Xerte, H5P allows users to create, share and reuse interactive HTML content in the browser. It provides functionality to create all types of learning objects that are available in Xerte, but also allows users to create interactive videos. This is an option that is not provided by Xerte.
H5P is not a standalone program like eXelearning (cfr. infra), nor is it a complete self-hosted PHP solution like Xerte. In contrast, H5P is only available as a plug-in for three PHP frameworks: Drupal, Wordpress and Moodle. Besides export to SCORM, H5P also supports xAPI tracking for some of their building blocks. At the time of writing, following resources can be tracked with xAPI: drag-and-drop, course presentation, interactive video, gap fill, labelling, multiple choice, summary, quiz and memory game.

- **Adapt** (‘Adapt’, 2016)

In its vision Adapt states the ambition to become a leading authoring tool for producing responsive, multi-device e-learning content. Adapt is driven by a community of enthusiast contributors.
Using Adapt an educator can create HTML5 learning objects with basic interactivity: gap filling, matching and multiple choice questions. There are no building blocks for other types of interactivity like drag-and-drop labelling, timeline exercises, Hangman or Memory.
Just like Xerte, the software needs to be installed on a web server. As of July 2017, Adapt has released a pre-alpha plug-in to enable xAPI tracking for Adapt learning activities. According to the documentation, at the time of writing, the plug-in still is not ready for production use.

- **eXeLearning** (‘eXeLearning’, 2018)

eXeLearning is an open-source authoring application to assist teachers and academics in the publishing of web content. The project started in 2007 by the University Of Auckland, New Zealand. In 2010 it was taken over by the Instituto de Tecnologías Educativas del Ministerio de Educación del Gobierno de España, mainly to make the transitions to the new emerging technologies at that time. Since 2013, eXeLearning 2.0 is available, providing functionality for export to both XHTML and HTML5 next to the old Flash export. eXeLearning supports both SCORM 1.2 and 2004 export. It is very user-friendly and nearly as powerful as Xerte. Compared to Xerte it has less possibilities in the variety of possible learning objects to create. eXeLearning is only available as a standalone program that has to be downloaded to the computer of the teacher, no web-based version is available.

- **RELOAD** (‘RELOAD Project’, 2008)

RELOAD is an acronym for Reusable eLearning Object Authoring and Delivery. The project is managed by the University of Bolton and aims to ‘facilitate the creation, sharing and reuse of learning objects and services’. In order to do so, it has developed a number of tools for teachers and educators. The project is Java-based and provides possibilities for export to SCORM 1.2 and SCORM 2004. The project has not been updated since 2008 but can still be used for Flash export. Hence, learning objects created with RELOAD cannot be used on mobile devices.

- **Learning Tools** (‘Learning Tools’, 2014)

Just like RELOAD, Learning Tools is an older project that has been abandoned since 2014. Until that moment, the project has been maintained by the University of British Columbia and was available for academic use. The project consists of a whole range of different tools and has some interesting unique learning objects like a handwriting tool and language pronunciation tool. Since this is an older, no longer maintained project, only Flash and no HTML5 export is provided.

3.3 Commercial Products

A large number of commercial products for creating learning objects is available on the market. For this thesis, we prefer to make use of free solutions, so we will not discuss these software packages in-depth. For the purpose of being complete, we will just mention some of the big players in this field at the moment.

All software mentioned here typically aims at companies to use their software to create e-learning courses. Some also have an academic plan. All these commercial products have the ability to export learning objects to HTML5 and provide xAPI tracking possibilities. Other features they all share are: immediate previews, export of learning objects to save to local files, a choice of course/quiz templates, video incorporation and drag-and-drop functionality.

Articulate Storyline 360 (‘Articulate Storyline 360’, 2018) is one of the industry leaders in authoring tools. They claim to be used by over 78.000 organizations in 151 countries. Besides the common functionality mentioned above, they also provide resources for import of PowerPoint presentation slides, webcam and screen recording, a survey tool and an image library of over 2 million assets to be used in learning objects. They also provide content creation training by experts in the form of webinars. Prices start from $999 for a yearly subscription.

Trivantis Lectora is another big name when it comes to authoring tools. With over 19 years of being in the business, several Fortune 500 and Global 2.000 Corporations rely on their software to create courses for training and certification of employees. Like Articulate Storyline 360 they provide an all-in-one solution, providing functionality for all the same extras that we mentioned for Articulate. At the time of writing (April 2018) the price for the most complete solution, Trivantis Lectora, is $2,595.

Smartbuilder (‘SmartBuilder’, n.d.) is one of the big players in this field, mostly used by corporations for their e-learning and in-house training. They provide the means to create custom learning objects for businesses, using a programming interface similar to Scratch. Therefore, creating learning objects in Smartbuilder requires more time and effort than using the building blocks of the before mentioned competitors. Smartbuilder also lacks a couple of features that Trivantis Lectora and Articulate Storybuilder 360 do have: screen and webcam recording, import from PowerPoint and image library to use when creating learning objects. Google and Cisco can be found among their clients. Prices start from $1.399 for a yearly subscription.

iSpring (‘iSpring’, 2017) is a bit of an exception in this list. With their iSpring Suite they provide the possibility to turn PowerPoint presentations into supercharged e-courses. Since PowerPoint is well known to most people, using it in combination with this software makes it quite easy to use it for the quick creation of learning objects. Even though all content creation happens in PowerPoint, all features available in the other tools mentioned are also available in iSpring Suite. Prices start from $720 for a yearly subscription.

Elucidat (‘Elucidat’, 2017) is another commercial authoring tool. Elucidat's claims to be used in 219 countries and counts some big names like The Open University, Tui and Tesco among its customers. They provide all features of their competitors, except for screen and web recording.

Finally, we also mention Gomo Learning (‘Gomo learning’, n.d.). It is used by companies like L’Oréal, British Airways, Deloitte, Banco Santander, Shell, Roche and the World Health Organisation to create in-house courses. In comparison with their competitors, their software lacks the functionality for importing PowerPoint, integrated screen and webcam recordings and survey tools. Prices start from $980 for a yearly subscription.

3.4 Conclusion

In the context of the TICKLE project learning objects need to be created. Rather than developing an own authoring tool for content creation, we have looked into different existing authoring tools in this chapter.

We have started with a definition and the main components of such an authoring tool. Then we have discussed the main features that all authoring tools have in common. Afterwards we have looked to free solutions as well as commercial products.

To avoid vendor lock-in and provide maximal control, we have decided to focus on open-source authoring tools. After comparing the different solutions, we have chosen to support the use of Xerte and H5P for content creation. Both authoring tools have some features in common that make them an acceptable choice for our purposes. They are both open source solutions and provide a wide range of possibilities for interactivity. In addition, learning objects created with Xerte and H5P can be exported in HTML5 format, which is essential for us since we plan to use the learning activities in a mobile 2.0 environment. Finally, both Xerte and H5P are driven by the enthusiasm of a large community and new versions are released on regular basis. Therefore, we don’t expect either of the projects to be abandoned soon.

4

Definition of the Student Profile

An essential part of the project consists of defining an appropriate student profile (RQ3). In this chapter we will identify the elements of this student profile. For this purpose, we start with reviewing related work on user profiling and how we can apply this to our system.

4.1 The User Model in Adaptive E-learning Systems

Over the last decade the fast evolving ICT has also entered the field of education. This (r)evolution has resulted in a lot of academic research on many different aspects of the best use of ICT for teaching.

One particular aspect that has been the subject of several articles in the academic literature is the adaptation of content presentation in mobile learning (m-learning) or electronic learning (e-learning) to the learner. This work on Adaptive E-learning Systems (AES) is very relevant to our project, since researchers in this field have made attempts to develop a so-called learner model or user model.

According to Brusilovsky and Millán, the learner model in an AES could incorporate a number of different aspects (Brusilovsky & Millán, 2007). This model has been widely accepted as a solid basis to create a user model in AESs. We will briefly discuss this user model here. The model is illustrated in Figure 4.

.

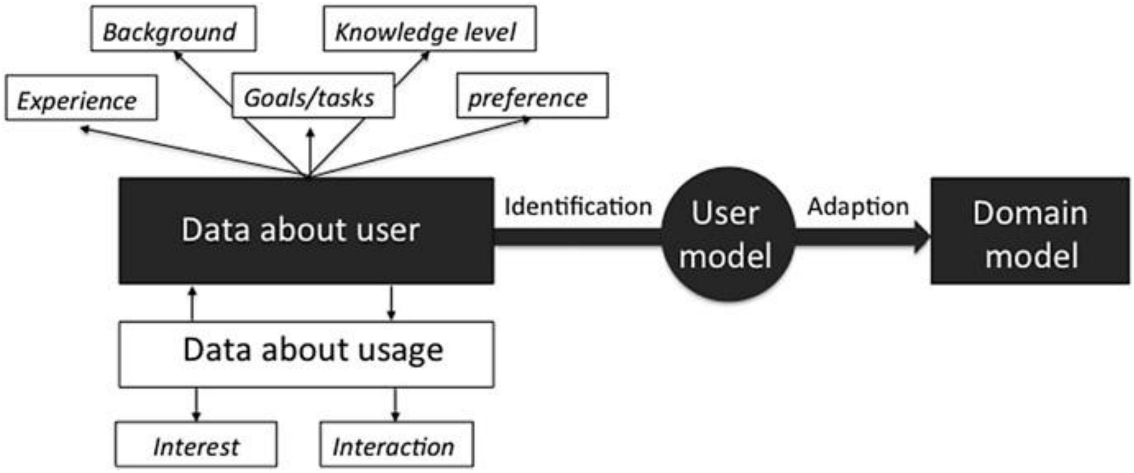


Figure 4 User Model. Reprinted from Brusilovsky & Millán (2007)

- **Knowledge**

The first aspect is the current knowledge level of the student. The actual knowledge of a student can be represented as an overlay model of the total domain knowledge of the course. This means that for each little part of the course, a Boolean value can be stored, representing whether the student has mastered this information. Knowledge level is incorporated in nearly all AESs.

- **User interest**

A similar overlay model can be used to model user interest. User interest modelling is more applicable to Adaptive Hypermedia Systems (AHS) then to AES. Examples of AHSs are systems that have a lot of information to offer to the user, e.g. encyclopaedia or museum guides. The model of the user's interest can be used to give recommendations to the user of what might be of interest to him or her. The user interest is often modelled by considering the collected data about usage and combining these data with a stored model of relations between information objects in the system. In this way, interest in one information object can increase the probability of interest in all related information objects.

- **Goal/tasks**

The goal of a learner is also important to know and is typically modelled by an overlay model. The system contains all possible goals and associates the current task with one of those goals. It is accepted in AESs that a learner can have only one goal at a given time. A so-called goal hierarchy can be kept within the system. It keeps track of the completion of lower-level goals to achieve higher-level goals.

- **Background**

The user's background refers to all previous knowledge related to the topic that a learner possesses. It is mainly used for adaptation purposes in AHSs and is not commonly used in AESs. Since the background doesn't change often, it is not modelled by an overlay model, but by a static model. Input about background into the system usually happens by the user himself or by a superior (teacher, administrator...).

- **Preference**

Individual traits or preferences contain all features of an individual's personality. In research on adaptive learning the main focus has been on cognitive styles and learning styles of learners, while personality traits (e.g. introvert/extravert) and cognitive factors (e.g. working memory span) have gotten much less attention. The cognitive style defines what representation of information the learner prefers, e.g. visual, written, and auditory. The learning style is defined as the preferred way of learning of a learner. This has gotten a lot of attention in numerous AESs. Both cognitive style and learning style can be determined by psychological tests or observation.

AESs or AHSs that store knowledge and/or goal into the learner profile mostly rely on a simple overlay model mapping what is done to what needs to be done. User interest can rely on a similar overlay model that needs to be updated according to the collected user data. As mentioned above, background is mostly stored in a static model.

In the next sections, we will elaborate on how each of the five types of user data of the model can be used to design a student profile for our system.

4.2 Knowledge

Unfortunately, the factor *knowledge* from the AES model discussed in the previous section does not prove to be very useful in the context of our project. The reason for this is that the main goal of our system is too different from the goals considered in m-learning and e-learning. Opposed to the main goal of most e-learning courses, the first aim of TICKLE is not to establish an increase of knowledge on a certain subject. In the TICKLE context there is no real course to be mastered by the student; the whole purpose of TICKLE is to reinforce the student's motivation for school and learning. Since we have no real course to be taught, in our case there will be no need for a domain model for representing the domain knowledge. Hence, there is also no need for an overlay model to represent the student's knowledge in a certain domain.

Although the main goal of our system is to increase the motivation of youngsters for learning, it is important to be aware of the user's general knowledge. Each learning activity might need some preliminary general knowledge to complete the activity. If the knowledge level of the learner is too low to deal with the activity, the youngster may not be motivated to start or finish the activity. A possible solution could be to include the required preliminary knowledge to complete the activity in the learning object.

The ICT knowledge of the student should also be considered. During the intake it is important to find out how familiar the youngster is with the use of computer and/or smartphone.

4.3 User Interest

The user interest is a factor that our software definitely needs to take into account. During the intake, the supervisor can explicitly ask the youngster for his/her interests, e.g. hobbies, music, film, games, books, pets, gardening .... Valuable information regarding the general user interest might also be collected via the parents, teachers and peers of the youngster.

Although the student is a potential dropout, he/she might have a preference for one or more subjects in school. These favourite school subjects should also be discussed with the coach during the intake. They can afterwards be checked with the teacher(s). The subject preference might be used during the selection of learning activities and the development of cards for the youngster. He/she will likely be more motivated for collecting cards and thus performing the associated learning tasks when they are in the favourite subject.

Another way to reveal user's interest is by collecting and analysing data about usage of the system. Examples of data that can be collect for this purpose are: the order the learner chooses to handle the learning activities, the time spend on an activity, the score on different (groups of) learning activities, ... The system could have an algorithm that will use the collected data to figure out the user's interests and make recommendations to the user, e.g. 'If you liked this learning experience, have a look at ... '.

A final way to reveal the user's interest is the use the overall usage data. The collected data from different users could be used to find matching patterns. If the data reveal that a lot of youngsters with a similar profile to the one of the user do well at a task, the system might suggest this task to the user. A possible way to get even more valuable data could be to end each test with a small rating concerning the liking of the learning activity (1 to 5 stars for instance).

A different kind of interest is the youngster's smartphone and related apps interest. A TICKLE report (Vlieghe, 2016) gives some interesting background information on this. According to this report, 94% of the youngsters use their smartphone daily. Most of the time this device is used for reading e-mails (60% daily), social media (60% daily) and SMS sending and receiving; Internet access is mostly done by using Wifi.

Most of the time, youngsters tend to be using a relatively small selection of apps: 54% of them has 10 to 20 apps installed, 27% has 20 to 40 apps installed. The smartphone is used for information gathering through websites, communication and entertainment. Communication happens on very regular basis via e-mail (91%) and social media; Facebook messenger (86%) and Snapchat (49%) being the most popular ones amongst youngsters. SMS is also used frequently. Smartphone use for entertainment consists mostly of music, video, games and social media. Also for entertainment, youngsters use Facebook (44%) and Snapchat (23%) more than Instagram (15%) and YouTube(15%).

From this information we can derive some type of data that might be useful in our student profile. It would be wise to include which social media the youngster uses frequently and is familiar with. That could then be the preferred way to notify him/her about learning activities. If users indicate to communicate a lot through SMS and e-mail, that might be an option too. This should also be stored in the student profile. Some more general information on smartphone use by the youngster can also be taken into account: how regular the device is used (daily, every other day, weekly,...) and for which purpose (information, news, wiki, social media, gaming, music, video...).

4.4 Background of the Target Group

The TICKLE project started with three studies to get the state-of-the-art on topics that would need to serve as background context for the later development.

The first of these three reports provides an overview of the existing academic literature related to early school leaving (ESL) (Vlieghe, 2014). It consists of two main parts: first, the prevalence of ESL in Europe, Belgium and Brussels, and second, the predictive value of factors that influence youngsters' decision of ESL.

In relation to the background information in our student profile, it may be very valuable to consider the factors mentioned in the second part of the study. These so-called influence factors have been split up in three groups: macro environmental or socio-demographic factors, micro environmental factors and individual factors.

In the group of macro environmental factors that are related to dropout and ESL we find gender, ethnicity, social-economic situation (SES), language proficiency and age. Although all factors are related to ESL and dropout, not all of them have a statistic predictive value by itself. For instance, relatively more male than female students dropout, but the gender by itself is not a statistic predictor. Ethnicity is also definitely related to ESL and dropout, but only has predictive value in combination with SES.

The strongest predictor of ESL and dropout is the SES of the student. An indication of the SES the youngster can be given by considering the family income, the parent's education and occupation. Besides SES, language proficiency also has large impact on school achievement and is therefore strongly related to ESL and dropout.

Age, like gender and ethnicity, is by itself no predictor, although a strong positive correlation between age and ESL is found: they both increase together.

For three of the five socio-demographic factors we discussed, there is at least a correlation with ESL. For the other two there is even a statistical causality found. Therefore it is advisable to take all five factors into consideration in our student profile.

As for the micro environmental factors, the study mentions three groups: family, school and peers. All three factors of the micro environment are very strongly related to ESL and dropout. They are also the easiest factors to alter: a little support from the micro environment can be the difference between dropping out or staying in school.

Several studies have shown that family has a significant influence on (possible) ESL. Important factors to consider here are the school involvement of the parents, as well as the expectations of the parents for their son/daughter.

The second group of the micro environment is the school. Especially the relation with and support from the teacher(s) is important here.

Finally there are the relationships with the peers that are important. Peers can be found in school (friends), out of school (other friends) and in the family (siblings). It is demonstrated in several studies that the relation with peers can be an important factor in relation to ESL.

More general, the whole context of the micro environment in which the students grows up should be considered. This includes the socio-economic, politic and cultural context of family, school and neighbourhood of the youngster. As many elements as possible should be included as background in the student profile.

Finally, besides macro and micro environment, there are also individual factors that need to be considered in relation to ESL. The individual factors that can be taken into account are school achievement and motivation. Both are strongly related to the ESL risk. The (lack of) individual engagement can be recorded in data about school attendance, misbehaviour (in school), (not) doing homework, the own education expectations and participation in school activities.

Defining the cultural context of a youngster is not straightforward. Culture is often defined as a kind of programming of the mind, leading to groups of people sharing the same preferences and values (Callahan, 2005). A group of people sharing the same culture will only have a similar thinking to some extent, but differences at the individual level may occur. Also, culture is not necessarily related to country boundaries.

Anthropologists have developed models for national cultures, determining the dimensions to allow cultural classifications (Hall, Hall, & others, 1989; Hofstede, 2003; Trompenaars & Hampden-Turner, 2011). Hofstede distinguished 5 dimensions to make classifications of national cultures. He researched the scores on each of these 5 dimensions in 74 countries, making it possible to get a general view on the cultural preferences of each nation.

We shall briefly discuss the 5 cultural dimensions from Hofstede, used for cultural classification. The first one is *power distance* (PD). Societies with a high PD have strong beliefs in hierarchy, while societies with a low PD put equality between all people first. The second dimension is *individualism* (IDV). In societies with high IDV people don't see themselves as part of the group in the first place. The individual is placed before the group. In countries with low individualism, people have a strong relation with the group they think to belong too. A third dimension from Hofstede is *masculinity* (MASC). Typical for high masculine nations is the high competition, while less masculine societies put caring for others first. *Uncertainty avoidance* (UA) is the fourth dimension. High UA indicates that people believe much in a very well structured and organized society, while less UA indicates the exact opposite. The last of Hofstede's dimensions is *long-term orientation* (LTO). Countries with a LTO strongly believe in tradition and invest in relations with others, while those with a short-term orientation mainly chase short-term goals.

The work of Hofstede has also been criticized. The main criticism is that the identification of culture is by nation and not by groups. But because Hofstede provides such tangible cultural factors and a lot of reference data, it has gotten a lot of attention, also in the field of user interface adaptation to culture. Academic work has resulted in a mapping between Hofstede's dimensions and User Interface design aspects, to adapt the interface as much as possible to match the cultural background of the user. Reinecke provides a good overview and summary of the outcomes of different studies in this field (Reinecke & Bernstein, 2013). A summary can be found in Table 2.

	Low Score	**High Score**
POWER DISTANCE	Different access and navigation possibilites; nonlinear navigation	Linear navigation, few links, minimize navigation possibilities
	Data does not have to be structured	Structured data
	Most information at interface level,	Little information at first level
	Friendly error messages suggesting how to proceed	Strict error messages
	Support is only rarely needed	Provide strong support with the help of wizards
	Websites often contain images showing the country's leader or	Images show people in their daily activities
INDIVIUALSIME	Traditional colors and images	Use color to encode information
	High image-to-text ratio	High text-to-image ratio
	High multimodality	Low multimodality
	Colorful interface	Monotonously colored interface
MASCULINITY	Little saturation, pastel colors	Highly contrasting, bright colors
	Allow for exploration and different paths to navigate	Restrict navigation possibilities
	Personal presentation of content and friendly communication with	Use encouraging words to communicate
UNCERTAINTY AVOIDANCE	Most information at interface level,	Organize information hierarchically
	Nonlinear navigation	Linear navigation paths / show the position of the user
	Code colors, typography & sound to maximize information	Use redundant cues to reduce ambiguity
LONG TERM ORGANIZATION	Reduced information density	Most information at interface leve
	Content highly structured into small units	Content can be arranged around a focal area

Table 2 Relationships between Hofstede's Dimensions and UI Design Aspects. Reprinted from Reinecke (2013).

Several studies have found other cultural aspects influencing human-computer interaction (HCI). So called cultural markers have been found, i.e. culturally specific design elements in websites (Barber, 1998). These include verbal attributes such as language and form (date, time, currency, printing format and measurement unit), visual attributes (images, color, text and layout) and audiovisual attributes (sound, animation, 3D).

It has been demonstrated that the writing system influences HCI (Chan & Bergen, 2005). Computer users with a left-to-right writing system tend to have the left part of the screen as their centre of attention. For people with a right-to-left writing system this is the other way around.

Another cultural aspect to consider is religion. It has been shown that Muslims prefer to use websites that are intended for people with the same religion over more neutral websites (Siala, O'Keefe, & Hone, 2004). Christians however didn't have this preference.

As for page layout Japanese tend to have a strong preference for clear headlines and bullet points, whereas Europeans prefer paragraph style (Cyr & Trevor-Smith, 2004). European and Asian sites also tend to use much more cultural specific symbols than US sites. In US and European sites the help function is much more prominent than in Japanese sites, but those contain a better index. Asians also have a stronger preference for symbolic navigation. There also is a considerable variation related to the use of colour. For example, Japanese use the colour red twice as much as Germans or Americans.

It has been demonstrated that colours have different symbolic meanings in different cultures (Chattopadhyay, Darke, & Gorn, 2002). In Islamitic countries the preferred colour would be green, whereas Christian tradition can be associated with red, blue, white and gold, and Buddhism with saffron yellow (Daniel, Oludele, Baguma, & Weide, 2011). Others have shown that the colours black, white, grey, blue and yellow are *international colours*, i.e. they are used and accepted in different cultures all over the world (Kondratova & Goldfarb, 2007) . The same study also gives country specific colour advice for use on websites (for 15 countries).

4.5 Goal/Tasks

The overall main goal of the TICKLE project is to contribute to the reduction of ESL and dropout of youngsters living in Brussels. This is the overall goal of the project. To achieve this goal two more low-level goals are considered.

In order to prevent school burnout, new media technologies will be used to encourage spontaneous learning. This spontaneous learning might have a positive effect on the self-confidence of the students and might also give a boost to their intrinsic motivation. Raising the self-confidence and intrinsic motivation of the student can both be considered as a first lower level goal.

The new media technology will consist of a number of small learning activities. This way of learning is also called micro learning (Gassler, Hug, & Glahn, 2004). The mere act of perfoming this micro learning on a regular basis can be considered as the second lower level goal. In the ideal scenario the execution of learning activities and collecting cards becomes part of the student's daily routing.

4.6 Preferences: Theoretical Background on Learning Styles

As discussed in section 4.1, it may be useful to consider personal traits and preferences of the student. Very often, the learning style of the student is considered and it may also be useful for our purpose. Therefore, first we should try to define what is meant by the term 'Learning style'. This is not as straightforward as one might think. A large number of researchers have defined 'Learning style' in their own way. Most of them however agree on the fact that the learning style is strongly related to the preferences of the learner to perceive and interact with the learning environment (Honey & Mumford, 1986; D. Kolb, 1981).

To model the learning style of a learner in an e-learning context, a number of possible solutions have been proposed. Very recent work has made an up-to-date overview of learning models used for these purposes (Doulik, Skoda, & Simonova, 2017). This article not only discusses the various models, but also attempts to categorize them by considering their conceptual overlap. We will limit ourselves here to briefly presenting the different learning style models that could be useful for our project.

- **The Dunn and Dunn learning style model** (Dunn, 2003)

The model proposed by Rita Dunn and Kenneth Dunn consists of five categories of so called stimuli that play a crucial role in the learning activity of an individual. The five stimuli are:

  - Environmental elements: sound, light temperature ...
  - Emotional elements: motivation, responsibility, persistence ...
  - Sociological elements: alone, pair, peer, group ...
  - Physiological elements: auditory, visual, tactual, mobility, intake of food, time of day...
  - Psychological elements: global (top-down) vs analytic approach (bottom-up)

This model belongs to the constitutionally-based learning styles. This conceptual approach accepts that all above mentioned learner's preferences are mostly fixed and won't change much over time. Tests to identify the preferences of a learner exist for different age categories ('Learning Styles', 2014).

- **Witkin's field dependence-independence** (Witkin & Goodenough, 1980)

Witkin distinguishes two kinds of learning preferences: field dependent learners and field independent learners.

The field dependent learner learns best with a top-down method, grabbing the global concept first and filling in the details later. They try to relate the study material to their own real world experiences. They require the goals to be set out for them and need reinforcement to keep them on track.

Field independent learners will use a more analytical approach. They will analyse each concept by itself without really considering the overlap. They don't do much effort to relate to their own experience. They are however very well organized and structured. They set out their own goals and need little reinforcement.

- **The Myers-Briggs model** (Myers, McCaulley, & Most, 1985)

The Myers-Briggs Type Indicater (MBTI) is a questionnaire to determine how people experience the world around them. MBTI finds its origins in the work of C.G. Jung, who stated that one of the four psychological functions is dominant over all other in their real-life experiences: sensing, feeling, thinking, and intuition.

With the test, the learning style of an individual can be determined. With a gradual scale, the test measures the scores of the learner for each of these functions:

  - Extravert/introvert
  - Sensing/intuitive
  - Thinking/feeling
  - Judging/perceiving

Depending on the score for each of these four criteria the preferred learning style will be determined on a 4x4 grid, containing a total of 16 possible learning styles.

- **Kolb´s Learning Style Inventory (LSI)** (D. A. Kolb, 1985)

In Kolb's theory, all knowledge is constructed from experience. Learning is merely the process of transforming this experience into knowledge. In his learning cycle, Kolb describes how this learning process takes place:

  - Concrete experience: a learner has an experience
  - Reflective observation: the learner reflects on the experience to determine if it matches with his expectations and understanding
  - Abstract conceptualization: this reflection may generate a new idea or an adaption to some prior constructed knowledge
  - Active experimentation: the learner uses this new constructed knowledge and tests it out to see if it is valid

Kolb also identifies four learning styles. This learning style of an individual is a product of one value on the x-axis (processing continuum) with one value on the y-axis (perception continuum). It is the product of how we do and feel when learning. A persons learning style can be placed in one of the four quadrants. See Figure 5 for an illustration.

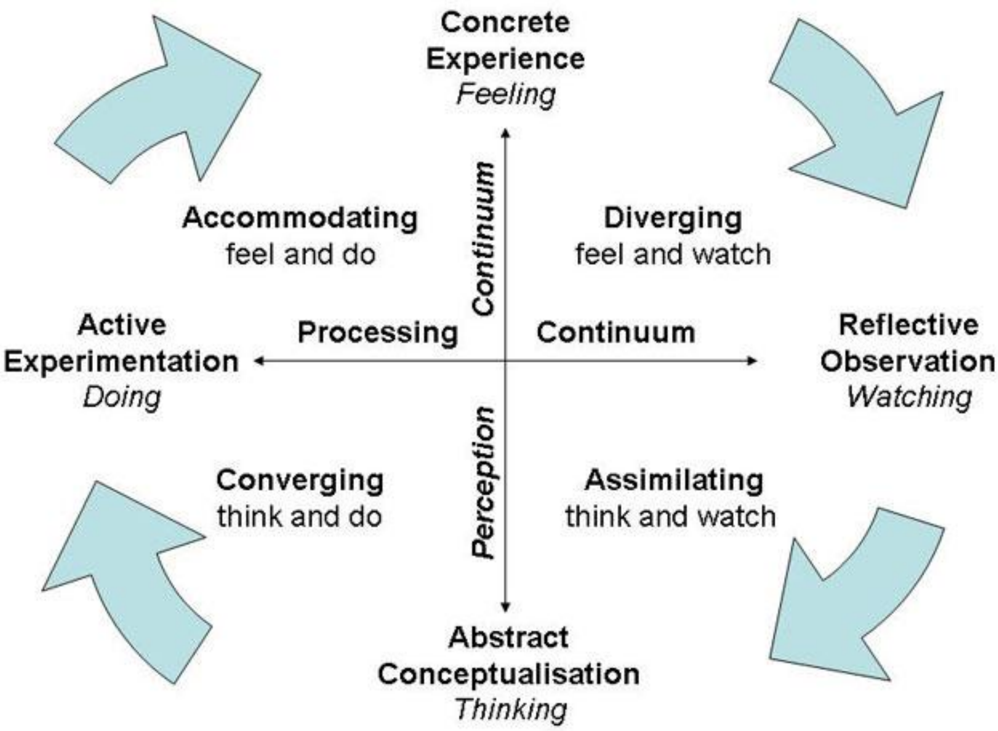


Figure 5 Learning Styles According to Kolb. Reprinted from Kolb (1985)

- **Honey and Mumford Learning Style Questionnaire (LSQ)** (Honey & Mumford, 1986)

Peter Honey and Alan Mumford elaborated on the work done by Kolb and distinguished four learning styles or preferences for individuals:

  - Activist: they learn best by doing. Activities that they can use to optimize their learning are brainstorming, puzzles, competition, group discussions...
  - Theorist: They are more analytical and learn in terms of models, concepts and facts. They derive their knowledge from understanding the theory behind what needs to be learned. They learn best using models, statistics, background information and stories.
  - Pragmatists: They learn best by trying out things in real world and see how they work out. They are experimenters; trying out the theories they are offered and see real world results. They can learn from discussions and case studies, but also from problem solving.
  - Reflector: A reflector observes what is happening and then thinks about it to figure it out. They take into account what they see, the measured data and other experiences they encounter. Learning may happen best in paired discussions, questionnaires, feedback or interviews.

- **The Felder-Silverman model** (Felder, Silverman, & others, 1988)
  Richard Felder and Linda Silverman have also developed a model for learning styles. It was originally designed with engineering students in mind. The model denotes four dimensions that contribute to determining the learning style of a learner. Each dimension is a continuum, where a learner is either oriented more or less towards one end of the dimension. The dimensions of personality that contribute to learning are:
  - Sensing – intuitive: how a learner prefers to take in or perceive information. Sensing learners tend to prefer concrete and practical (facts) learning, while intuitive learners prefer concepts, theories and concepts.
  - Visual – verbal: denotes the way the information should be presented for optimal learning. Visual learners prefer presentations, pictures and diagrams, while verbal learners will prefer writing and oral means of learning.
  - Active – reflective: how information is processed by a learner. Active learners will try out while learning, while reflective learners rather think about the concepts.
  - Sequential – global: the path one prefers towards grasping the concept of the information presented. Sequential learners take little, linear steps to construct new knowledge, while global learners think more top-down.

- **The VARK test (Fleming, 1995)**

The VARK test is developed to help learners identify their individual learning preference. After getting the result from the VARK test, a learner can apply study techniques that match best with his/her learning style. It helps learner

VARK is an acronym for the four types of learning styles in this model:

- V(isual)
- A(ural)
- R(ead/Write)
- K(inesthetic

Visual learners have a preference for viewing information in images, diagrams, charts, graphs... They are advised to convert learning materials to graphics and to use color.

Aural learners prefer to hear the information or speak aloud what needs to be retained. This includes learning from presentations, watching tutorials, group discussions and debates.

Learners with a Read/Write preference learn best using words in one form or another: Powerpoint slides, the Web, books, journals, wikis or lists. Flash cards (vocabulary or information) with words or word diagrams tend to work best for these kind of learners.

Kinesthetic learners prefer to learn by experience, movement, touching and doing things. They learn best by getting hands-on practice and personal experience. This includes learning by demonstrations, simulations, role-play or field trips.

4.7 Preferences: 'Measuring' the learning style

In order to determine the learning style of a student, different solutions have been proposed for the models that have been discussed in the previous section. The approaches for 'measuring' the learning style we will present in this section were proposed in the context of adaptive learning, each one using some theoretical model as background.

For the Dunn and Dunn learning style, a survey with 118 questions has been developed in the IWeaver project (Wolf, 2003). The result of the questionnaire consists of scores for the five preferences in the perceptual domain and the four preferences in the psychological domain.

One of the first tests to determine the learning style using the Felder-Silverman model was a survey consisting of 28 questions in the Computer Systems course CS383. The test results in a score for each of the model's four measurable dimensions. Since the beginning of this century the de facto standard for measuring the learning style according to the Felder-Silverman model has been the Felder-Soloman ILS questionnaire (Felder & Soloman, 1999). This survey has been used in numerous projects in the field of adaptive learning. The TANGOW system (Paredes & Rodríguez, 2002) focused on the sensing/intuitive dimension and WHURLE only used the test score on the verbal/visual axis (Brown, Brailsford, Fisher, Moore, & Ashman, 2006). More recent work has combined the results from the ILS questionnaire with a short questionnaire on the preferred learning materials (.ppt, quiz, video, audio, text, ...) to complete the learner profile (Abdullah, Daffa, Bashmail, Alzahrani, & Sadik, 2015). Other recent work has identified the Felder-Silverman learning style preferences as a suitable model for e-learning (Jegatha Deborah, Baskaran, & Kannan, 2014).

In the Adaptive Educational Hypermedia System (AEHS) INSPIRE (Papanikolaou, Grigoriadou, Kornilakis, & Magoulas, 2001), the LSQ developed by Honey & Mumford (Honey & Mumford, 2006) was used to determine the learner's preferences. My Online Teacher (MOT) has been an ongoing project for years (Cristea & Calvi, 2003) and uses Kolb's questionnaire to determine the learning style of a student, mainly to distinguish between converger-oriented and diverger-oriented students.

The OPen Adaptive Learning Environment (OPAL) was one of the first adaptive learning projects that consisted of a database of SCORM objects (Conlan, Dagger, & Wade, 2002). OPAL makes use of the VARK test.

In more recent work, the MBTI test has also served as a basis for constructing the user profile of an adaptive learning system. Kim conducted research in the context of a LISP course that students had to take (J. Kim, Lee, & Ryu, 2013) after they had completed the MBTI questionnaire. The resulting preferences (E-I, S-N, T-F and J-P) were stored in the user profile.

AEC-ES is an adaptive instructional system that takes the field dependency (FD) or field independency (FI) of the learner into account (Triantafillou, Pomportsis, & Demetriadis, 2003). Fourth year undergraduate students were presented Witkin's GEFT to determine their FD and FI score. A similar experiment was conducted by Mitchell (Mitchell, Chen, & Macredie, 2004).

4.8 Learning style preference conclusions

Just like in most AESs, user preferences should be included. We will have to store the learner's cognitive style and/or learning style. In this chapter we have thoroughly discussed the learning styles that are used in many different adaptive systems.

We have seen that all conceptual learning styles have an accompanying questionnaire, enabling to determine the learning style of the user in the model at hand. Our target group however consists of students (with a high risk of) dropping out. Since it is already very hard to get them motivated for school, it seems infeasible to let them complete a questionnaire to determine their learning preferences.

Since we would like to have a minimal idea of learning preferences into the user profile, we suggest to rely on the input of the teacher or coach here. After all, the teacher (from class room experience) and coach (from the intake and student tracking system of the school) may have enough information to determine his or her learning preferences.

Because we are not using questionnaires and solely rely on the teacher's and coach's observations, the conceptual learning model we use should be as simple as possible. The first model we therefore propose is Witkin's FI/FD. From classroom observations it might be possible for a teacher to determine if a student learns best with top-down or bottom-up approach. Another model that may be used without the need for a questionnaire is the VARK model. With this model, just like with the model from Witkin it might be possible for a teacher to tell whether a student's learning preference is visual, aural, read/write or kinaesthetic.

4.9 Other Information

One of the three context studies for the TICKLE project point out that 97% of the Flemish houses has access to Internet, 98% of them via broadband Wifi (Vlieghe, 2016). Only 36% has an additional data subscription. Although the Internet access percentage for Brussels houses is lower (90%), it can still be considered fairly high. In the report, other interesting numbers on devices can be found: 93% of the Belgian houses have a computer and in 98% of the Belgian households there is at least one smartphone.

Although the report shows that devices and Internet connection access are fairly widespread, it is necessary to check during intake whether or not the youngster owns a mobile. Since TICKLE aims to target mainly mobile devices, this seems a necessary requirement. Possible access to other devices like tablets, laptops and computers can also be inserted in the user profile. For those devices we could also add if these are for personal use or shared among others. The report by Vlieghe mentions that all other devices expect the smartphone, are mostly shared by people living under the same roof.

If we want the TICKLE project to be strongly related to the city of Brussels and use a map to let the user collect cards in various city places, it must also be recorded in the student profile whether the youngster has sufficient access to Internet. Therefore, the student profile must contain information regarding data subscription and homespot.

4.10 Conclusions

Based on the various studies in the previous sections, we have identified a list of factors that can or will be useful for our student's profile:

General information

- First name
- Last name
- school ID
- date of birth(→ age)
- e-mail
- gender
- nationality

School (social)

- school attendance
- misbehavior
- school activities
- expectations

School achievement

- test scores (good and bad subjects)
- grades
- homework

Educational stability

- mobility (between schools)
- dropout history

Educational attainment

- years completed
- diploma received (e.g. lower secondary)

Personal

- attitudes

- motivation
- autonomy
- competence
- learning style
    - Witkin's FD/FI
    - VARK

Socio-demographic background (macro level)

- gender
- ethnicity
- SES
    - family income
    - education parents
    - occupation parents
- education language proficiency
- age

Socio-demographic background (micro level)

- family
    - parent school involvement
    - parent expectations
    - other important family members (besides siblings, e.g. grandparents, stepfather/mothers...)
- school
    - teachers with good teacher-student relation
- peers
    - in school
    - out of school (hobbies, other friends,...)
    - siblings
- neighbourhood
- cultural background
    - nationality

  - (first) language spoken → with mother, father, siblings, friends
  - writing system (left to right or vice versa)
  - Content to avoid
  - page layout
  - colours
  - metaphor
  - images

User interest (topical)

- favourite subjects at school
- hobbies, including sports
- music (genre, bands)
- film (genre, films)
- General interest: other(e.g. gardening, pets,health,history...)
- collect data:
  - crowdsourcing → in learning activities
  - (star) rating → in learning activities
  - like→ in learning activities

User interest (smartphone)

- use of smartphone: regularity (daily, every other day, weekly, ...), purpose (information, news, wiki,..)
- use of e-mail
- use of SMS
- use of FB Messenger
- use of Snapchat
- use of Instagram
- use of YouTube
- Facebook
- music
- video
- gaming

Goals

- level 3: avoid dropout or ESL
- level 2: self-confidence goes up, intrinsic motivation goes up
- level 1: learning activities completed, cards collected

Device and data requirements

- smartphone device (personal use)
- computer (personal use)
- Internet access: wifi, data subscription, homespot

5

Learning Object Metadata

The learning objects created with Xerte and H5P will have to be stored and retrieved. For each learning object metadata needs to be stored, in order to provide easy search and filter functionalities and to make them shareable with others, even outside the scope of this project. Best is to use a standard for this. In this chapter we discuss the possibilities and we elaborate on the chosen standard.

5.1 Metadata

Since the late 1990's learning objects have found their way into the field of education. Several different repositories were initiated, each one using its own database and own description of learning objects for indexing and cataloguing. The data used for this purpose is often referred to as metatags or metadata, i.e. data about the data. A machine should be able to read these metadata and compile the information for a human user, for instance in order to discover learning objects according to his/her interest.

With the increasing popularity of e-learning, the need arose to create a standard for these metadata to enhance the interoperability of learning objects between different repositories. Metadata can be defined as structured data about data. Metadata contain descriptive information about the objects in a catalogue. The best-known real-life example of metadata is a library card catalogue.

5.2 Important Standards for (Learning Object) Metadata

In the early 2000's some important standards have been set up. We discuss them in the following sections.

5.2.1 Dublin Core Metadata Initiative (Weibel, 1999)

One of the first large initiatives for developing a metadata standard was the Dublin Core Metadata Initiative (DCMI). It was mainly developed to facilitate discovery of content by machines. Although the Dublin Core standard did not focus specifically on metadata for learning objects, it can very well be used for it. The *unqualified* Dublin Core only contains 15 data elements to describe the data of an object. They are chosen deliberately broad enough to be applicable in a wide range of domains. Dublin Core can also be used with *qualifiers* to further refine the 15 main data elements. DCMI has added an own controlled vocabulary, but gives the freedom to the user to use other vocabularies.

5.2.2 IEEE LOM

The IEEE LOM standard has used concepts from earlier attempts by different organisations (McClelland, 2003). Since the late 1990's IMS (Instructional Management Systems Learning Consortium) had been the driving force for setting out standards for the major learning course management systems like Blackboard. On the other hand, some large learning object repositories had entered the field, each using their own standards. In 1998, IMS and Ariadne, one of Europe's major learning object repositories, have joined forces and submitted a proposal to the IEEE which would be the foundation for the LOM standard. This approach was largely based on earlier work done in this field by the Dublin Core Group. In an appendix the IEEE LOM provides a mapping between its own metadata set and the Dublin Core, implicating that the DCMI was used for IEEE LOM as underlying principle.

As defined in its final draft 1484.12.1-2002 (IMS Global Learning Consortium, 2006), the LOM consists of nine categories, which are often referred to as the *base schema*: General Characteristics, Lifecycle Characteristics, Meta Metadata Characteristics, Technical Characteristics, Educational Characteristics, Rights Characteristics, Relation Characteristics, Annotation Characteristics and Classification Characteristics. Each of these nine categories contains numbered items for further refining the metadata. For instance, the General category (1) contains an identifier (1.1), a title (1.2), a language (1.3) ... In total the IEEE LOM uses seventy-seven possible metadata items for the description of a learning object. The main structure is depicted in figure 6.

The IEEE LOM provides two ways for binding, i.e. expressing the elements of the LOM standard through a formal language or syntax. IEEE LOM provides bindings for XML (P1484.12.3) and RDF (P1484.12.4). The practical undertake for creating RDF bindings has been halted and the mapping between IEEE LOM and Dublin Core has been proposed for RDF representations. In more recent work the first steps in creating an ontology for IEEE LOM have been undertaken (Casali, Deco, Romano, & Tomé, 2013).

However, up to this day most large repositories still use metadata in XML format, mainly stored in relational databases and generated on demand.

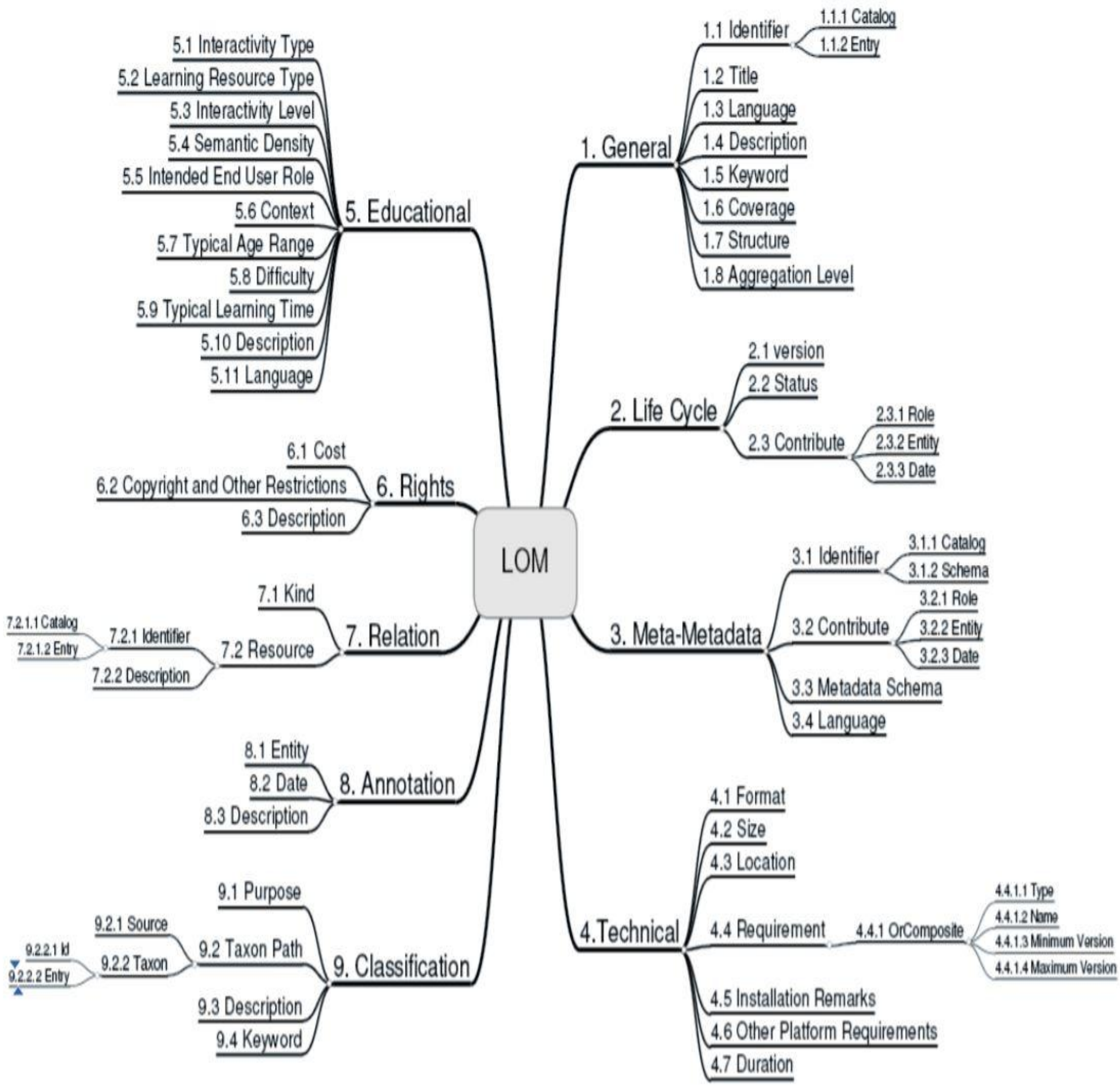


Figure 6 IEEE LOM Specifications Overview. Reprinted from Casali, Deco, Romano, & Tomé (2013)

5.3 Learning Object Taxonomies

In a repository the learning objects need to be organized in a certain way. Taxonomies are used for this. We could think of taxonomy as a tree with different information nodes. When one goes up the tree in the direction of the root, the information becomes more general, while going down in the tree the information will be more specific.

Since one of the main objectives of metadata is to facilitate the retrieval of learning objects a user is searching for, metadata should be organized in a way that a search can come up with the most relevant results. For this reason, the IEEE LOM contains a special category for the classification of a learning object in a given taxonomy.

Several approaches are used for adding keywords to the metadata (Lehman, 2007). The first one is called 'full text indexing' and is done programmatically. The program extracts all substantives from the learning object and adds them to the keyword field. The second approach uses an ontology, typically representing the learning object's topic with a taxonomy. A user can then search by category, subcategory or any combinations of all these using Boolean operators AND and OR. Most of the time, this will yield much more relevant results then using 'full text indexing', since the keywords that are used for search are the same keywords used for the representation of the learning object.

Besides the *Classification* element, there is also a *General.Keyword element* (LOM 1.5). For these keywords, no specialized controlled vocabulary is used. The keyword tags can be freely chosen by the user.

5.4 Controlled Vocabularies

The main purpose of using a standard for describing the metadata of a learning object is to attain interoperability with other systems. Interoperability in this context means that other systems should be able to *understand* what a learning object is about, merely by the description of its metadata. In order to facilitate the understanding of the semantics of the description, most repositories use a controlled vocabulary. For certain elements of the metadata, only a value from the controlled vocabulary can be chosen. Usually a controlled vocabulary comes with an extensive documentation, explaining all possible values and their exact semantics. The use of a controlled vocabulary for the values in the key/value pairs in a learning object’s metadata ensures that the description can hardly be misinterpreted by different users of these metadata. The IEEE LOM has its own controlled vocabulary for certain elements, the LOMv1.

5.5 Application Profiles

After the IEEE LOM standard had been set, various organizations found the need to customize the metadata even further. For this purpose application profiles have been setup. Such an application profile defines its own set of metadata, coming from different available schemas. An application profile also contains documentation on how it should be used. Application profiles are often referred to as *cores*.

Cancore is one such an application profile that has become very popular for learning object metadata (Friesen, Roberts, & Fisher, 2002). It claims to be fully compatible with IEEE LOM, but has reduced the number of fields to only thirty-six, organized in nine groups. In the UK an attempt was made to reduce the data elements in the IEEE LOM to a bare minimum. This application profile, known as the UK LOM Core (Barker, 2005), reduces the number of data elements to 18, leaving 23 others as optional. Finally, we will here also mention the SingCORE, which was developed by Singapore eLearing Framework (SeLF) (Ismail, Yin, Theng, Goh, & Lim, 2003). Studies have found that in the field of learning objects a very large part of the application profiles is based on the IEEE LOM standard, while for other fields Dublin Core is the most popular standard (Malta & Baptista, 2014).

In the next paragraph we will discuss a more recent and widely accepted application profile, the Learning Resource Exchange Application Profile.

5.6 Learning Resource Exchange Metadata Application Profile

European Schoolnet is a project funded by the European Union to facilitate portability of learning objects to schools all across Europe (David Massart, 2009). The website contains a large database of learning objects, most of them available in different languages.

In October 2011, European Schoolnet released their final version 4.7 of the Learning Resource Exchange Metadata Application Profile (LREMAP) (D. Massart, Shulman, & Van Assche, 2011). The document describes in detail how learning object metadata should be organized for allowing exchange and discovery in the Learning Resource Exchange (LRE) for Schools, i.e. European Schoolnet's repository of learning objects.

In the next parts of this section we will discuss this LREMAP (Learning Resource Exchange Metadata Application Profile) and indicate how it has been used for our purpose. First, we will present the two main parts of this application profile: ILOX and LOM. Next, we will discuss the general outline and main structure of LREMAP. Since LREMAP does not use all elements that are defined in IEEE LOM, we will then elaborate on the selection of IEEE LOM elements that have become part of this application profile. Not all IEEE LOM elements are mandatory in LREMAP, so for our project we had to choose which optional elements we would incorporate and which ones could be left out. This is presented in the next part, together with a detailed overview of all IEEE LOM elements we have selected for our TICKLE project. In the final part of this chapter we will explain how we have used the chosen standard to enable a mapping between user interest and learning object topics.

5.6.1 The Metadata: ILOX + LOM

For description of the metadata, the organization uses an application profile that combines the IEEE LOM standard with IMS LODE Information for Learning Object Exchange (ILOX) specification, a framework for organizing existing standards such as LOM. For the values in a learning object's description a controlled vocabulary is used. The LRE application profile makes use of Vocabulary Bank for Education ('Vocabulary Bank for Education', 2017).

Describing the metadata of a learning object according to these specifications, allows a learning object to be found and exchanged between different organizations. The LRE repository contains learning objects from different origins: museums, commercial publishers, non-commercial publishers, ministries of education... Since all of them are using the same proposed application profile, they are machine readable and can thus be accessed by the LRE repository.

5.6.2 ILOX: The General Outline

The ILOX can be seen as the main data structure of the metadata (D. Massart et al., 2011). ILOX describes a learning object at four different levels: *work*, *expression*, *manifestation* and *item*. *Work* is the most general description of the object, i.e. the learning object in its most *raw* form. It contains an identifier and a description. *Expression* describes the different versions of that learning object. It can for instance have an English and a Dutch version. Each of those is considered an *expression*. Every version can be available in different formats, e.g. HTML5, Flash, SCORM ... In this application profile these are called *manifestations*. Finally, at the lowest level there are 'tangible' copies of the learning object, available at a certain location and identified by a URI. These are the *items* of the learning object. The metaphor for the four levels of object description is well depicted in Figure 7.

WORK Expression Manifestation Item

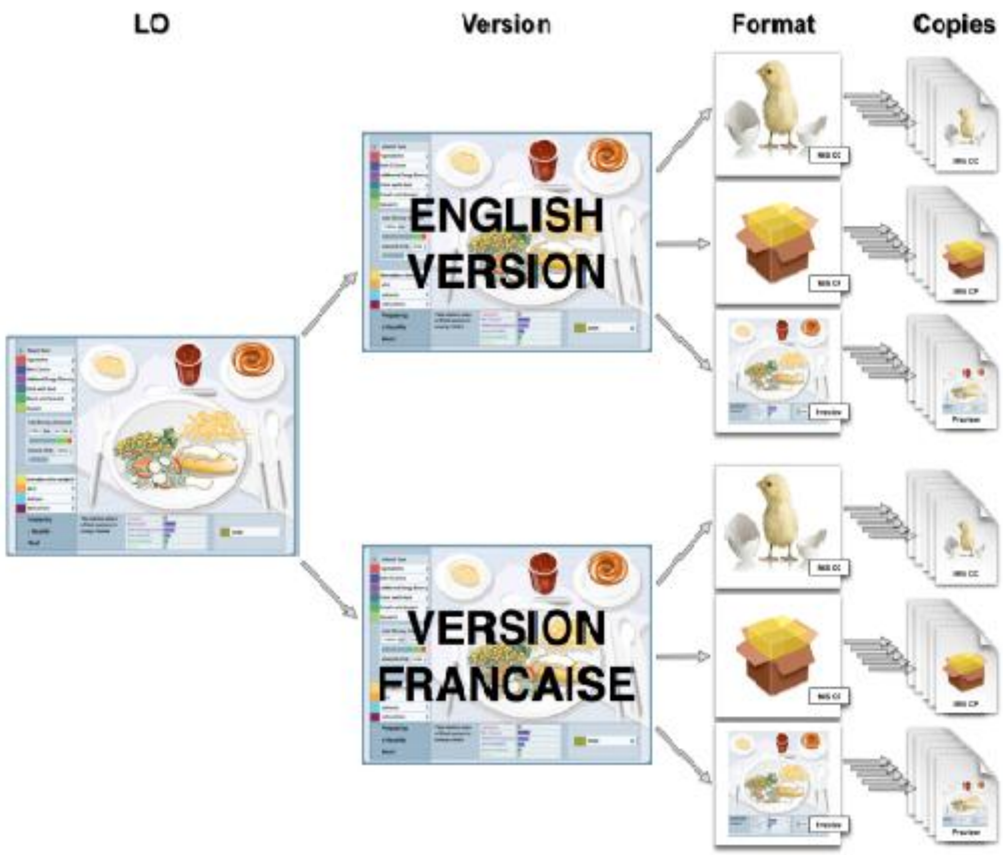


Figure 7 ILOX Description Levels of a Learning Object.Adapted from D. Massart et al.(2011)

This is how the general outline of a learning object description looks like:

```
<work>
        <metadata>
        <!-- Here comes the IEEE LOM description of the metadata -->
        </metadata>
        <expression>
                <manifestation>
                        <item>
                        </item>
                </manifestation>
        </expression>
</work>
```

5.6.3 LOM

The metadata elements: mandatory, recommended or optional

The metadata description of a learning object can be done at the level of work, expression, manifestation or item and is done by the IEEE LOM standard. This standard has been chosen to provide semantic interoperability of learning objects. Between the ILOX metadata tags, the elements of the IEEE LOM are available for use to describe the learning object itself.

For the values used in the description, the application profile only allows the controlled vocabulary from the Vocabulary Bank for Education (VBE). To encode controlled vocabulary, both ILOX and IEEE LOM use a similar approach, i.e. a two-element type where the first element references the used vocabulary and the second contains the actual value.

For description of the metadata not all 77 items of the IEEE LOM have the same status: there are elements that are mandatory, others are recommended and the rest of them are optional. Only five are mandatory: General.language (LOM 1.3), Educational.Learing Resource Type (LOM 5.2), Rights.Cost (LOM 6.1), Rights.Copyright and Other Restrictions (LOM 6.2) and Rights.Description (LOM 6.3).

Thirteen other IEEE LOM elements are recommended. The thirteen recommended elements are: General.Title (LOM 1.2), General.Description (LOM 1.4), General.Keyword (LOM 1.5), General.Coverage (LOM 1.6), LifeCycle.Contribute (LOM 2.3), LifeCycle.Contribute.Role (LOM 2.3.1), LifeCycle.Contribute.Entity (LOM 2.3.2), LifeCycle.Contribute.Date (LOM 2.3.3), Meta-Metadata.Language (LOM 3.4), Educational.Intended End User Role (LOM 5.5), Educational.Learning Context (LOM 5.6), Educational.Typical Age Range (LOM 5.7), and Classification (LOM 9). A number of those are advised very strongly.

Besides the five mandatory and thirteen recommended LOM elements, the LRE Metadata Application Profile also supports all other IEEE LOM elements as optional. From the remaining 59 IEEE LOM elements, only these seven are mentioned in the documentation: General.Structure (LOM 1.7), Meta-Metadata.Contribute (LOM 3.2), Meta-Metadata.Contribute.Role (LOM 3.2.1), Meta-Metadata.Contribute.Entity (LOM 3.2.2), Meta-Metadata.Contribute.Date (LOM 3.2.3), Educational.Typical Learning Time (LOM 5.9), Educational.Description (LOM 5.10).

The LRE Thesaurus for hierarchical classification

For the *Classification* (LOM 9), European Schoolnet strongly advises to use the LRE Thesaurus they provide. This thesaurus contains a (controlled) vocabulary to describe the subject of learning objects. The LRE Thesaurus is part of the VBE, where the most up to date version can be accessed. This thesaurus supports multi-language subject classification. At the moment of writing the available languages are Danish, Czech, German, Greek, English, Spanish, Finnish, Hebrew, Hungarian, Italian, Dutch, Polish and Swedish.

A very interesting feature of the LRE Thesaurus is that it doesn't just provide a taxonomy that is applicable in a school context, but also defines relations between terms. The relations between terms that are defined in the LRE Thesaurus are *Broader Term* (BT), *Narrower Term* (NT), *Use* (from a non-preferred term to its preferred term), *Use for* (from a preferred term to its non-preferred term(s)), *Related Term* (RT), *Linguistic equivalent* (LE).

The LRE Thesaursus' starting point consists of seventeen *Top Terms* (TT). They form the root of the taxonomy tree structure and are chosen to be best suitable as indexing starting points for learning objects in a school context. These seventeen Top Terms are: Communication/information/document, Culture, Health/safety/handicap, Individual development, Facilities/equipment/materials, Countries and geopolitical areas, Learning/research, Society, Environment, Modern languages, Teaching/training/evaluation/guidance, Content of education, Educational system, Organisations, Political/social/interpersonal relations/integration/segregation, School activities, Leisure activities.

5.6.4 Using the Learning Resource Exchange Metadata Application Profile for TICKLE

For the TICKLE project we propose to use the Learning Resource Exchange Metadata application profile. This application profile was established with interoperability in different languages between European countries in mind. The use of ILOX and IEEE LOM has now become widely accepted, which makes it a good choice for the representation of metadata in the field of learning objects.

When implementing the chosen application profile a number of choices should be made. Important decisions need to be made on which elements from the IEEE LOM should be included in the metadata. Since our chosen application profile only contains five mandatory elements, we should decide which of the remaining elements from IEEE LOM we will include. It has been well established that the completeness of the metadata has an effect on the reuse of learning objects (Zervas & Sampson, 2014a). Teachers looking for learning objects will be much more eager to choose those objects with complete metadata descriptions.

On the other hand, several studies have also found that authors of learning objects are hardly willing to spend any extra time to enter the metadata into the system (Duval & Hodgins, 2003). To address this issue, the semi-automatic generation of metadata has been proposed. The burden of entering metadata would be alleviated if the system would present the teacher with prefilled fields in the editor for entering metadata. These suggestions could be harvested from parsing the content of the learning object itself, metadata from related learning objects and metatags used by the author in other learning objects. It should however always be up to the teacher to confirm or reject these suggestions.

With our selection of metadata from the IEEE LOM we have tried to find a good balance between including sufficient metadata to facilitate reuse and making it feasible for authors to complete the metadata of a learning object. We have of course included all five mandatory elements and nearly all recommended elements have also been added to the selection. For reuse and discovery of learning objects we find it important to have most elements of LOM.5 (Educational) available in the metadata.

We have also included a number of elements that can be easily generated by the system. Since an author is logged in when adding a learning object to the repository, it is fairly easy to add the name of the author, the date and the identifier of the learning object without interference of the teacher.

Finally, we attach a great deal of importance to the Classification element (LOM 9). One of the reasons for choosing the Learning Resource Exchange Metadata application profile is the attached LRE Thesaurus as taxonomy for learning objects. We treat this as required fields, since they will be of great importance for the discovery and retrieval of learning objects according to a user's interest.

In our selection we have identified a total of 28 elements of metadata to be completed. We believe that at least fourteen of those can be semi-automated by the system. Those elements would only need confirmation by the author about the prefilled values in the editor. This is possible for fields like Catalog and Entry, which together represent a unique identifier that is generated by the system. System generated data can also be proposed for elements like Role and Entity, since the author is logged in at the moment of creation and a controlled vocabulary is used for the Role. Everything that has to do with Rights (LOM6) will also be standard input for the metadata, since an 'open' license will be used and all content will be free of cost. Finally, the Date will be added to the metadata in an automated way.

The other half of the metadata will require some attention of the author. This will be necessary for all properties that are used to describe the learning object's semantics. All elements representing the Educational context of the learning object (LOM 5) cannot be generated by a system. Adding the correct tags for a learning object from the LRE Thesaurus will also be a task for the author. Since having an object labelled with the correct tags is of utmost importance for reuse and discovery of learning objects, this is a task that will require human insight. The system could merely propose some tags, but only the author will be able to choose the right ones from the over 3000 available labels in the LRE Thesaurus.

Besides the tagging system from the LRE Thesaurus, we have deliberately chosen to give authors the possibility to add their own tags in the form of keywords (LOM 1.5). Recent studies have shown that this kind of *social tagging* or *folksonomies* can have a positive influence on enrichments of the description of learning objects in repositories (Zervas & Sampson, 2014b). There have been testimonials of user tagging that have found their way into the controlled vocabulary of the system they were added to.

The table in the next section shows our selection of elements from the IEEE LOM, together with an explanation, data type for representation and an example for each element.

5.6.5 TICKLE Selection from the IEEE LOM Set

Nr	Name	Explanation	Datatype	Example
1	**General**	This category groups the general information that describes this learning object as a whole.		
1.1	**Identifier**	A globally unique label that identifies this learning object.		
1.1.1	**Catalog**	The name or designator of the identification or cataloguing scheme for this entry. A namespace scheme.	CharacterString	ISBN, TICKLE
1.1.2	**Entry**	The value of the identifier within the identification or cataloguing scheme that designates or identifies this learning object. A namespace specific string.	CharacterString	LO2368
1.2	**Title**	Name given to this learning object.	LangString	("en", "The life and works of Leonardo da Vinci")
1.3	**Language**	The primary human language or languages used within this learning object to communicate to the intended user.	CharacterString (use a 2 letter code from ISO 639-1., else ISO 639-2)	'en-GB', 'fr'
1.4	**Description**	A textual description of the content of this learning object.	LangString	("en", "This is a collection of animal sounds recorded in conifer

				forest at different seasons")
1.5	**Keyword**	A keyword or phrase describing the topic of this learning object.	LangString	("en", "animal sounds") ("en", "lynx")
2	**Life Cycle**	This category describes the history and current state of this learning object and those entities that have affected this learning object during its evolution.		
2.3	**Contribute**	Those entities (i.e., people, organizations) that have contributed to the state of this learning object during its life cycle (e.g., creation, edits, publication).		
2.3.1	**Role**	Kind of contribution.	VocabularyTerm - Source: VBE vocabulary "LRE.roleValues" (16 values)	author, publisher
2.3.2	**Entity**	The identification of and information about entities (i.e., people, organizations) contributing to this learning object. The entities shall be ordered as most relevant first.	CharacterString- vCard, as defined by IMC vCard 3.0 (RFC 2425, RFC 2426).	<![CDATA[BEGIN:VCARD VERSION:3.0 N:Massart;David;;; FN:David Massart ORG:European Schoolnet END:VCARD]]>
2.3.3	**Date**	The date of the contribution.	DateTime	2017-10-26

3	**Meta-metadata**	This category describes this metadata record itself. This category describes how the metadata instance can be identified, who created this metadata instance, how, when, and with what references.		
3.1	**Identifier**	A globally unique label that identifies this metadata record.		
3.1.1	**Catalog**	The name or designator of the identification or cataloguing scheme for this entry. A namespace scheme.	CharacterString	ARIADNE
3.1.2	**Entry**	The value of the identifier within the identification or cataloguing scheme that designates or identifies this learning object. A namespace specific string.	CharacterString	LRE2003
3.2	**Contribute**	Those entities (i.e., people, organizations) that have affected the state of this metadata during its life cycle (e.g., creation, validation).		
3.2.1	**Role**	Kind of contribution.	VocabularyTerm - Source: VBE vocabulary "LRE.roleMetaValues (4)	creator, enricher,...

3.2.2	**Entity**	The identification of and information about entities (i.e., people, organizations) contributing to this metadata. The entities shall be ordered as most relevant first.	CharacterString- vCard, as defined by IMC vCard 3.0 (RFC 2425, RFC 2426).	see 2.3.2
3.23	**Date**	The date of the contribution.	DateTime	2017-10-26
5	**Educational**	This category describes the key educational or pedagogic characteristics of this learning object.		
5.2	**Learning Resource Type**	Specific kind of learning object, The most prominent kind first.	VocabularyTerm - Source: VBE vocabulary “LRE.learningResourceTypeVal ues” (27)	drill and practice, educational game…
5.3	**Interactivity Level**	The degree of interactivity characterizing this learning object. Interactivity in this context refers to the degree to which the learner is supposed to take an active part in dealing with the learning object.	VocabularyTerm - Source: VBE vocabulary “interactivityLevelValues” (5 values)	very low … high
5.5	**Intended End User Role**	Role of principal user(s) for which this learning object was designed, most prominent first.	VocabularyTerm - Source: VBE vocabulary “LRE.intendedEndUserRoleVal ues (7)	learner

5.6	**Context**	The principal environment within which the learning and use of this learning object is intended to take place.	VocabularyTerm - Source: VBE vocabulary "LRE.contextValues" (12)	compulsory education
5.7	**Typical Age Range**	Age of the typical intended user.	LangString - Language: Must use at least langString with "x-t-lre". Other langString codes are allowed when "x-t-lre" is present. String: Typical Age Range is expressed as a range Minimum-Maximum age in years	("x-t-lre", "10-12") ("x-t-lre", "7-U") ("x-t-lre", "U-12") and ('en-GB", "U-12")
5.8	**Difficulty**	How hard it is to work with or through this learning object for the typical intended target audience.	VocabularyTerm - Source: VBE vocabulary "difficultyValues"	very easy ... very difficult
5.11	**Language**	The human language(s) used by the typical intended user of this learning object.	CharacterString (use a 2 letter code from ISO 639-1., else ISO 639-2)	'nl, 'fr'
6	**Rights**	This category describes the intellectual property rights and conditions of use for this learning object.		

6.1	**Cost**	Whether use of this learning object requires payment.	VocabularyTerm - Source: VBE vocabulary "costValues"	no, yes
6.2	**Copyright and Other Restrictions**	Whether copyright or other restrictions apply to the use of this learning object.	VocabularyTerm - Source: VBE vocabulary "copyrightAndOtherRestrictions Values"	no, yes
6.3	**Description**	Comments on the conditions of use of this learning object.	Mandatory if 6.2 equals "yes" NOTE: Only one description per language.	("en", "See copyright notice)
9	**Classification**	This category describes where this learning object falls within a particular classification system.		
9.1	**Purpose**	The facets taken into account for classifying this learning object.	VocabularyTerm - Source: VBE vocabulary "purposeValues" (9)	discipline, idea, prerequisite

9.2	**Taxon Path**	A taxonomic path in a specific classification system. Each succeeding level is a refinement in the definition of the preceding level.	When element 9.1: Purpose equals 'discipline', this element is mandatory and should be use to store LRE Thesaurus descriptors.	
9.2.1	**Source**	The name of the classification system.	LangString - This data element may use any recognized "official" taxonomy, or any user-defined taxonomy. When element '9.1 Purpose' equals 'discipline', the value of this element must be ("x-none", "LRE-0001"), which corresponds to the VBE identifier of the LRE thesaurus.	("x-none", "LRE-0001")
9.2.2	**Taxon**	A particular term within a taxonomy. A taxon is a node that has a defined label or term. A taxon may also have an alphanumeric designation or identifier for standardized reference. Either or both the label and the entry may be used to designate a particular taxon.		

9.2.2.1	**Id**	The identifier of the taxon, such as a number or letter combination provided by the source of the taxonomy.	CharacterString - When element '9.1 Purpose' equals 'discipline', this element is used to store the VBE identifier of a LRE Thesaurus term.	1441,255,895
9.2.2.2	**Entry**	The textual label of the taxon.	LangString - When element '9.1 Purpose' equals 'discipline', this element can be used to store one or more translations of the LRE Thesaurus term identified in element 9.2.2.1.	(nl,'Cultuur')

Table 3 TICKLE Selection from the IEEE LOM Set. Adapted from D. Massart et al. (2011)

5.7 Mapping Learning Objects to User Interest

5.7.1 Approaches for ranking learning objects

In practice, most of the time we expect that a teacher or coach will create a learning object and card with a student or group of students in mind. In these cases the teacher may be familiar with the student(s) and can consult the data stored in the student's profile. Since the system will try to enhance the reusability of both learning objects and (collections of) cards, the application should be able to assist the teacher/coach in the search and discovery of learning objects and cards that are related to the interests of the student. Hence, the system will have to contain a ranking algorithm for discovering the learning objects suitable for the student and his interest and ranking them by relevancy. This will facilitate teachers and coaches in selecting appropriate learning objects for a student. In the literature different approaches for a relevancy ranking system of learning objects have been proposed. In this paragraph we will give a quick summary of some relevant studies in this field.

There exist two main older approaches that are still commonly used today. The first approach is used by MERLOT and is based on manual rating (Kumar et al., 2007). Learning objects are reviewed and rated by a group of people that are considered experts in the field and/or by the users of the learning objects. For each search the results show the learning objects ordered by their rating. The main problem with this approach is that a lot of learning objects never receive a rating and get penalized for this in the ranking. This implies that some good quality and relevant learning objects will not show up on top of the rankings, just because they were not peer-reviewed.

The second approach, used by ARIADNE, requires no human intervention. The system composes a vector by executing a full text scan of the metadata. In the same manner a vector is composed from the query string. The algorithm then uses a metric to calculate the distance between both vectors, returning the results in descending order. Several metrics can be applied for this, but TF-IDF still remains one of the most used ('tf–idf', n.d.). Main problem with this approach is that users hardly take the time to add a full and meaningful description in the metadata (Friesen, 2004). Therefore measuring the correlation between what is in the metadata and the query results in poor rankings.

More recent work has tried to include more parameters to come up with better rankings. Contextualized Attention Metadata (CAM), representing the user's interaction with the system can be recorded and taken into account (Ochoa & Duval, 2006). Based on the learning objects a user has entered into the system or searched for, the labels or tags a user has used and the number of downloads, a graph about the user's interest can be constructed. This graph can then be used for making recommendations.

Several other studies have examined other possibilities of building a user profile by collecting data, also in distributed e-learning environments (Dolog, Henze, Nejdl, & Sintek, 2004). Although we might expect the personalized rankings to outperform the text-based rankings, it has been demonstrated that text-based rankings are often more accurate than ranking based on automatic collected user data.

We would like to mention a final approach for ranking learning objects here. Researchers have proposed to use the same taxonomy for the student profile and the learning objects (Calle, 2007). It might be expected that this will yield better results for relevancy rankings. The drawbacks mentioned in this context are that it takes extra work to complete the student profile and the user is bound to the taxonomy to choose from. Also, this ranking system will only work when the exact same taxonomy is used for both the student profile and the learning object.

5.7.2 **Representing user interest**

For revealing learning objects that might suit a student's interest, we will use the last approach mentioned in the former paragraph. As mentioned before, we will be using the LRE Thesaurus as taxonomy for the learning objects. An author will have to label each learning object with one or more LRE tags when adding it to the repository.

We propose to also use this same taxonomy to describe the student's interest. Several studies have used quite comparable taxonomies for building a User Interest Hierarchy (H. R. Kim & Chan, 2003). Others have used the Open Directory Project ('Open Directory Project', 2018) categories, also known as *DMOZ*, for research on web search personalization (Sieg, Mobasher, & Burke, 2007) and interest based personalization search (Ma, Pant, & Sheng, 2007). In this last study, a network was trained to provide a mapping of the user interest to the categories used in the Open Directory Project. The ODP was the largest and most comprehensive Web directory, maintained by a community of volunteer editors. Others have also proposed to use the ODP or Yahoo categories to model user interest (Chen, Zhao, Yu, & Wan, 2010; Chirita, Nejdl, Paiu, & Kohlschütter, 2005; Han, Shen, Miao, & Luo, 2010; White, Bailey, & Chen, 2009). Before closing in March 2017, DMOZ totalled over one million categories and subcategories.

The cited studies confirm that the ODP categories have been used in academic literature to represent user interest. Since the enormous amount of categories, it wouldn't be feasible to use all these for describing the user interest in the student profile. Just like earlier studies (Ma et al., 2007), we will try to determine if there is a possible mapping between the ODP top categories and the topics in the LRE Thesaurus. Table 4 provides such a possible mapping:

ODP main categories	LRE Thesaurus topics
Arts	Art
Business	Business Management
Computers	Computer
Games	Video Game – Computer Game
Health	Health
Home	Family – Food – Home economics
News	Magazine – Newspaper – (choose news topic)
Recreation	Leisure
Reference	School
Regional	Country
Science	Science
Shopping	Home economics
Society	Society
Sports	Sport

Table 4 Mapping ODP Categories with LRE Thesaurus Topics

DMoz also provides special categories for *Teens and Kids*, which is exactly our target audience. When comparing these kids categories with the main categories in table above, we may notice that not all main categories are present in the *Kids and Teens* categories. Vice versa, a small number of extra categories, especially suitable for kids and teenagers, have been added. We have also compared these to find a mapping with the topics in the LRE Thesaurus topics. The possible mappings that were established can be found in Table 5.

ODP Kids and Teens categories	LRE Thesaurus topics
Directories -> Animals	Animal
Directories -> Politics	Politics
Directories -> Homework	Homework
Directories -> Teens Only	Teenager
Entertainment -> Actors and Actresses	actor
Entertainment -> Bands and Artists	artist - Music
Entertainment -> People -> Athletes	athletics
Entertainment -> People -> Authors	author
Pre-school	pre-school education
School time	School
Teen Life	Teenager

Table 5 Mappings of the Kids and Teens Categories with LRE Thesaurus Topics

5.7.3 Informal description of the TICKLE mapping algorithm

In the last two sections we have proposed to use the terms from the LRE Thesaurus for both tagging the learning objects and the student interest. A GUI called Tag Explorer has been designed to aid the author in finding the right tags. Since both are using the same controlled vocabulary, we might expect to discover learning objects that are relevant to the student's interest. The algorithm used to reveal learning objects according to a student's interests will be described here in an informal way.

In order to provide a ranking, we compare the tags used for the learning objects with the student interest tags. The more matching tags are discovered, the higher a learning object will rank. Within the tags, we make distinction between tags from the controlled vocabulary (LRE Thesaurus) and the keyword tags added by the author. Since the keywords are given by the author, we expect them to be more specific. Therefore they will get a weighing factor that ranks them higher than exact match tags from the controlled vocabulary.

Since the LRE Thesaurus contains the relations between terms, we also have the possibility to search for matches with related terms, when no exact match tags are found. It is clear that an exact match outweighs any related-topic matches. In Table 6 we have summarized the possibilities for the matching of the tags, with a weight assigned in descending order. With all the matches found, a weighted vector can be constructed to determine a 'matching value'. The higher a match is found in the table, the more weight it brings to the vector. The higher the total matching value of the vector, the higher a learning object will rank.

	Controlled Vocabulary Tags	Keyword(s)
22	more than one exact match	more than one exact match
21	one exact match	more than one exact match
20	more than one exact match	one exact match
19	one exact match	one exact match
18	more than one related topic match	more than one exact match
17	one related topic match	more than one exact match
16	more than one related topic match	one exact match
15	one related topic match	one exact match
14	no exact or related match	more than one exact match
13	no exact or related match	one exact match
12	more than one exact match	more than one related topic match
11	one exact match	more than one related topic match
10	more than one exact match	one related topic match
9	one exact match	one related topic match
8	more than one related topic match	more than one related topic match
7	one related topic match	more than one related topic match
6	more than one related topic match	one related topic match
5	one related topic match	one related topic match
4	no exact or related match	more than one related topic match
3	no exact or related match	one related topic match
2	more than one related topic match	no exact or related match
1	one related topic match	no exact or related match
0	no exact or related match	no exact or related match

Table 6 Informal Description of the Matching Algorithm

5.7.4 **The main categories**

For searching or tagging learning objects, the LRE Thesaurus uses some concrete starting points. When trying to discover learning objects by subject, the following main topic categories are used for the search: art, astronomy, biology, chemistry, citizenship, classical languages, cross-curricular education, culture, economics, educational administration, environmental education, ethics, European studies, foreign language, geography, geology, health education, history, home economics, informatics/ICT, law, mathematics, media education, music, natural sciences, philosophy, physical education, physics, politics, pre-school education, primary education, psychology, religion, school-community relationship, social sciences, special (needs) education, technology.

We have looked at the search/discovery features in some of the well-known learning object repositories: Merlot ('MERLOT II', 2018), , LeMill, Connexion ('OpenStax CNX', 2018), Mit Open Courseware ('MIT OpenCourseWare', 2018) and Oasis Commonwealth of Learning ('OAsis', 2018). Table 5 shows the comparison in top categories between the different learning object repositories.

When comparing the top categories of these repositories with those in the LRE for Schools, we noticed that the granularity for search at the top level in *LRE for Schools* is much finer then in the other repositories. On top of that, the refinement in subcategories is also much more refined than in most other repositories. LRE for Schools also provides extra relations between search-terms, an option we did not find in the six compared repositories. For each term, the LRE provides a list of associated broader terms and narrower terms.

This fine grained starting point for the top terms, together with the well-defined relations between terms, make the LRE for Schools an excellent choice for tagging learning objects and user interest in our TICKLE project.

MERLOT	ARIADNE	LEMILL	CONNEXION	MIT OC	OASIS COL	LRE for Schools
9 top disciplines: -Academic Support Services -Arts -Business - Education - Humanities - Mathematics and Statistics -Science and Technology - Social Sciences - Workforce Development Hierarchy: - at most 5 levels deep - mostly 3 levels deep	Unavailable at the time of writing	26 top categories: -Art -Biology -Chemistry -Citizenship -Cross-curricular education -Culture -Economics -Educational administration -Environmental education -Ethics -Foreign languages -Geography -History -Informatics or ICT -Language and literature -Mathematics -Media education -Music -Natural sciences -Philosophy -Physical education -Physics -Politics -Psychology -Religion -School-community relationship	6 top categories: -Arts -Business -Humanities -Mathematics and Statistics -Science and Technology -Social Sciences Hierarchy: -No subcategories -just keyword search per category	11 top categories: -Business -Energy -Engineering -Fine Arts -Health and Medicine -Humanities -Mathematics -Science -Social Science -Society -Teaching and Education Hierarchy: -At most 3 levels deep -Subtopics at level 2 - Specifications at level 3	10 top categories: -Education -Engineering -Health Sciences -Languages -Livelihoods -Mathematics -Natural Sciences -Skills and Trades -Social Sciences -Technology Hierarchy: - just 10 main topics -no subcategories	41 top categories: -art -astronomy -biology -chemistry -citizenship -classical languages -cross-curricular -education -culture -economics -educational administration -environmental education -ethics -European studies -foreign language -geography -geology -health education -history -home economics -informatics/ICT -law -mathematics -media education -music -natural sciences -philosophy -physical education -physics -politics -pre-school education -primary education -psychology

						-religion -school-community relationship -social sciences -special (needs) education -technology Hierarchy: - different levels of subcategories (broader terms, narrower terms) - related terms

Table 7 Search by Category in the Seven Repositories

5.8 Conclusion

In this chapter we have presented different important standards for (learning object) metadata. We found that the Dublin Core Metadata Initiative has been the leading metadata standard for general data. Extending this Dublin Core, the IEEE LOM was identified as the most widespread standard for metadata of learning objects. In the IEEE LOM, a taxonomy tag can be used for the organization of learning objects in a repository.

Application profiles or cores define their own metadata. After discussing different cores that have been used in the field of education, we have elaborated on the Learning Resource Exchange Metadata Application Profile (LREMAP). LREMAP consists of a combination of the ILOX specification and an IEEE LOM subset to describe the metadata of a learning object. The ILOX specification describes the high-level characteristics, while the IEEE LOM subset of elements contains the low-level metadata.

After choosing LREMAP as the application profile for the TICKLE metadata, we have identified the selection of IEEE LOM elements that are most suitable for our purpose. We have then proposed to use the LRE Thesaurus as the controlled vocabulary for the taxonomy tags of a learning object, as well as for the user interest. The use of this controlled vocabulary will enable us to map the learning objects to user interest. Finally, we have discussed the main categories of the LRE Thesaurus and compared them with the main categories of some well known learning object repositories.

6

Requirements

In this section we will give an overview of the requirements the authoring tool should meet. Since the tool will be part of the TICKLE project, it is necessary to first consider the general aims of TICKLE to understand the specific context where the authoring tool will be used. Therefore we will first describe the general aims of TICKLE and the relation of the authoring tool to other parts of the project.

After understanding the more general context, we will describe the requirements for the authoring tool. We will make a distinction between functional and non-functional requirements. After a more general description of the requirements, a numbered list of requirements will be given to ease referring to them.

To have a better understanding of the functionality of our system, a use-case diagram has been developed and the use-case scenarios have been created to describe the required workflow in more detail.

6.1 Context of the Authoring Tool

6.1.1 General Aims and Context of TICKLE

The TICKLE project aims to use modern technologies such as web 2.0 and social media to increase the learning engagement of youngsters in Brussels and ultimately avoid school dropout. In order to assist teachers and coaches (called authors in the rest of the document) in using modern technology to keep youngster motivated, a playful environment will be created to reactivate youngsters with the ultimate aim to prevent school dropout.

The application development has been split up in three main parts. One of those three project parts is the subject of this master thesis. To provide the context of this thesis, we will here briefly discuss the relation of our topic to the other parts of the TICKLE project.

First of all, TICKLE should have an authoring environment where authors can create challenges. A *challenge* in this context is a short interactive learning activity that has been scheduled for a certain student. Once such a challenge is created, it is ready to be performed by a student. These short attractive challenges should enhance the learning motivation for youngsters in our target group.

As of section 6.2 this chapter will elaborate on the requirements to create and manage these challenges, which is the subject of this thesis. In the remainder of this section, the parts of the TICKLE project that are outside the scope of this thesis will be briefly discussed. Although not part of our work, it is important to have an understanding of how the authoring tool will be connected to the other parts.

6.1.2 **The city card Environment**

The challenges created by the application presented in this thesis are not standing alone. They will be integrated in a strong motivational environment: i.e. the city card environment.

Therefore, authors will attach their created challenges to certain city locations. Each challenge created by the teacher can be executed at (or near to) a specified location. Students will see a city map on their mobile device containing all locations in the city where a challenge is waiting to be completed by them.

Furthermore, each time a challenge is successfully completed, the student will be rewarded with the digital card that is containing the challenge. The cards also need to be created by authors. Cards may belong to one or more collection of cards. Collections are used to group cards, e.g. based on the topic, for instance *Music*. It might be expected that the student will be more motivated to perform a challenge if he wants to collect all cards in a certain collection within his or her field of interest.

Therefore, the city card environment contains two parts: the city card authoring environment and the delivery environment. In the city card authoring environment, authors will be able to create cards and card collections and connect them to locations. Creating cards is done by completing card templates. This will mostly take place on a laptop or PC. Attaching the locations to the cards could be done through a mobile environment.

Once challenges have been created and added to cards, the cards should be made available to the student. This will be done by the delivery environment. Note that the learning activities mentioned in the title of the thesis refer to what we have called challenges.

The delivery infrastructure will also need to keep track of completed challenges and will send reminders for learning activities that are still available. This may be done through social media and using proven persuasive techniques to improve the response of the youngsters.

6.1.3 Learning Analytics Module

Finally, a last module for providing learning analytics will be developed.

The goal of this module is to provide teachers and coaches inside in the activities performed by a youngster and to provide authors insight in the take on and performances of challenges. Therefore, this module needs access to the student results on the challenges performed. The learning analytics module also aims at discovering certain patterns for success or failure of challenges by students.

With the outcome of this module, adaptations for optimizing the card delivery process used by TICKLE can be made. The learning analytics could lead to changes in the way of targeting youngsters, the subjects of the learning activities, the type of challenges used for certain students, …

6.1.4 The TICKLE System Architecture

To clarify how the different modules we have discussed above are connected, we present the complete TICKLE system architecture in this paragraph. The diagram representing the system architecture can be found in Figure 8.

The authoring tool will have two types of users that will interact with the system:

- Administrators/supervisors: they handle the user management for a school, organization or a number of schools, i.e. they provide access to the system for teachers and coaches (i.e. the authors) by creating accounts for them. Besides this user management they have all rights that a regular author has.
- Authors: the authoring tool will mostly be used by teachers and coaches to create challenges for youngsters.

Besides the users we just mentioned, the system will also interact with two external modules:

- The external authoring tool: the Xerte or H5P authoring tool to create learning objects and export them to allow importing them into the system.
- The city card authoring environment: here the created challenges will be linked to cards that have been created and associated with locations in the city.

As for data storage, the authoring tool will have its own database to store all information about student profiles, learning objects metadata and challenges. For now, the application will also have read access to the Learning Record Store (LRS) to check the status of the challenges. Once the development of the learning analytics environment will start, this part of the functionality will have to be transferred to that module. For the time being, a sort of mini learning analytics module has been integrated into the authoring tool to get a quick overview of the status of challenges.

A general overview of the TICKLE architecture at the highest level can be found in Figure 8. As reflected in this structure diagram, our authoring environment is located within the supervisor/author back-end and will communicate with the city card authoring environment through REST web services. This implicates that the city card authoring module can be hosted in a different physical environment than our application.

The two external learning object authoring tools supported at this moment, Xerte and H5P, will be hosted on the same server as our authoring tool. This is not only a logical choice since both systems are closely connected, but it also avoids a number of problems with cross origin request errors that will arise when hosting both on different servers.

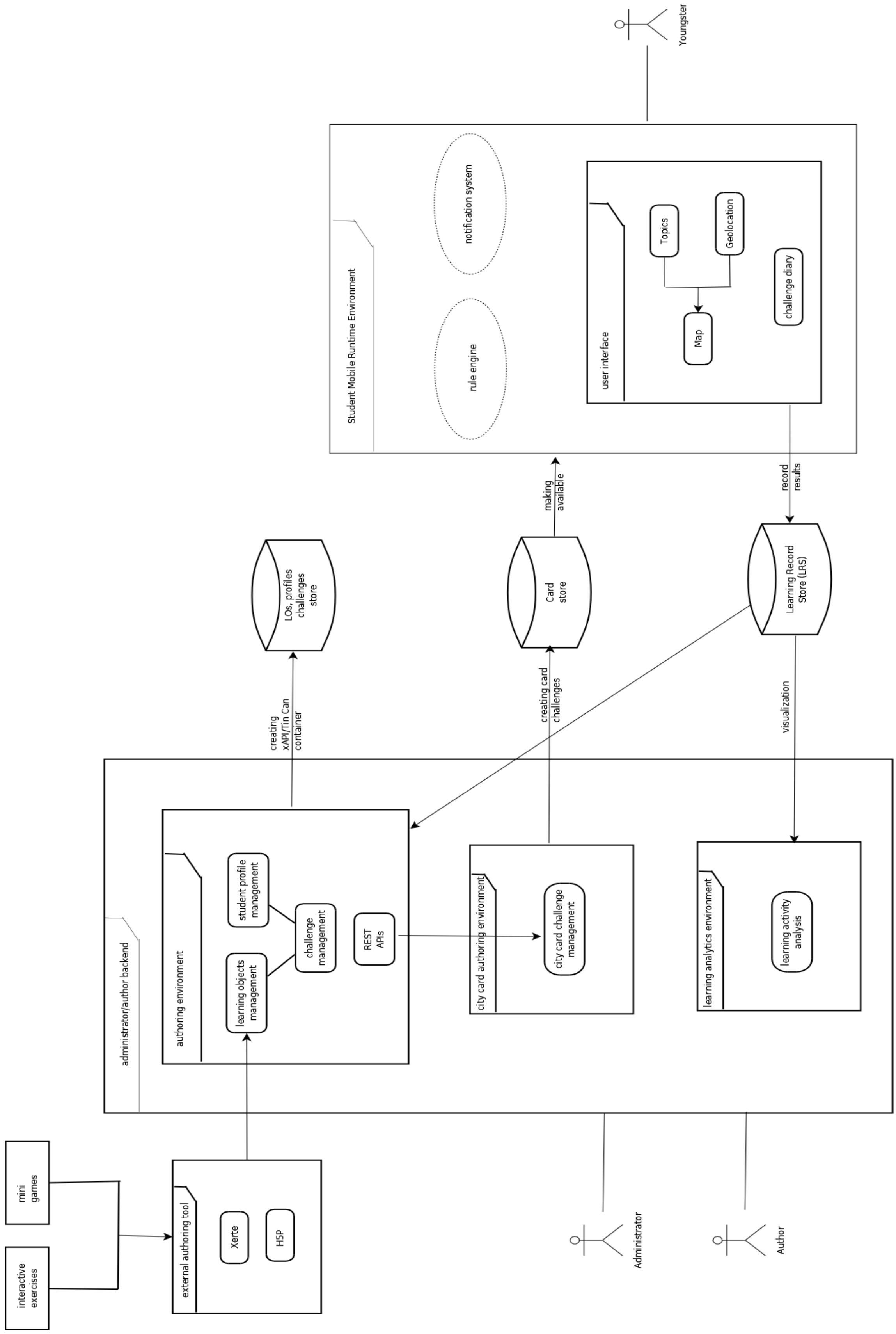


Figure 8 The TICKLE System Architecture

6.2 Requirements for the Authoring Environment

6.2.1 Functional Requirements

The authoring environment will have two types of users: teachers/coaches (authors) and supervisors. A user with the role of supervisor can be seen as the person responsible for the TICKLE project in a group of schools or an organization dealing with school dropout. A user with the role of teacher/coach is a youngster's teacher/coach.

Supervisors will be responsible for the user management of the application. They will have rights to create, view, edit and (de)activate user accounts for authors. Authors will be able to login to the application with the credentials provided by the supervisor. Besides the user management, the supervisor will have exactly the same rights as an author. This will provide maximum flexibility for participating schools and organizations.

The key feature our authoring environment should provide is creating and managing *challenges*. We refer to a challenge as a learning activity that has been scheduled for a student. This means that our authoring tool should allow the creation of learning objects or learning activities. In the remainder of this text, we will use the terms learning object and learning activity as synonyms.

The types of learning objects we aim for are short interactive exercises, sometimes referred to as educational widgets. Often they present some basic information to the student, followed by a learning activity that contains interactive and/or gaming elements. These *fun* aspects can come in the form of well known games like Memory and Hangman, or interactive exercises like matching pairs, multiple choice questions or drag-and-drop exercises. All these types of learning activities should be more appealing to youngsters than classical learning assignments or content. We will not give an explicit list of game types or interactivity types to be supported as part of the requirements, but from the short informal description in this paragraph it should be clear what type of learning activities we are aiming for.

Furthermore, we intend to use these learning objects in a mobile web 2.0 environment, since the delivery environment will be a mobile application. This implicates that the learning objects must be in a mobile friendly format or should have the possibility to be exported into a format suitable for our purposes.

Once a learning activity is created, the system must provide a means to save it for later use and reuse. Therefore, we need a so-called learning objects library. In this library all created learning objects will be stored. To enhance reusability of learning objects by other authors, we propose to give all authors in the system access to all learning objects for use in their challenges.

Because this learning objects library could grow very quickly, searching for suitable learning objects could become harsh. Therefore, it will be important that the system facilitates searching the learning objects library. Users should be able to find the desired learning object quickly, even if the system contains a large number of learning objects.

To accomplish easier search and better targeting, each learning object that is added to the system must also contain metadata that fully describe the learning object. The metadata description should contain all possible information that could be useful to authors when searching for learning objects with certain characteristics. Moreover, to facilitate reusability even outside the TICKLE system, the learning object's metadata should also be made available in compliance to a current metadata learning standard.

Finally, the author of a learning object should have the possibility to make changes to the learning object's metadata. Functionality for viewing, editing and deleting a learning object in the learning objects library should be provided by the system.

Challenges will always be created for students. Therefore, the application will need to contain a student management module. Since we will be working in the specific context of early school leaving, the student profile should contain as much relevant information as possible for this context. The student profile must be a valuable piece of information for authors wanting to get better insights into a student's situation and for matching existing learning objects to students.

It should also be possible to view, edit or delete the student profiles that have been added. As with learning objects, the number of student profiles available might become too large to easily locate the student profile one is searching for. Therefore, the system should provide a means to facilitate search in the student profiles.

Adding student profiles and learning objects to the system are the necessary steps to be taken in order to reach the main goal of the authoring tool: the creation of challenges for students. Challenges are always created for a particular student. Each challenge will contain only one learning object from the learning objects library.

Note that, as discussed above, in the aims and context of this thesis, the challenges created in this authoring tool will be used in the city card authoring environment. Therefore, the authoring tool should provide access to this extern module via APIs. Since all input of student profiles and learning objects will happen through our authoring tool, only read/retrieve functionality for these APIs need to be provided. For the learning objects both the learning object and the according metadata should be accessible. An author should however be able to create challenges from within the city card authoring environment, for instance if he finds himself at a certain location and decides *on-the-fly* to create a challenge with a matching learning object from the library. Therefore our authoring tool will provide a web service access point where the extern module can retrieve and create challenges via APIs.

Some basic results of challenges that are performed by students will be recorded by the system. As discussed in the section above, a separate learning analytics module will be developed. Since this is outside the scope of this thesis, our authoring tool will only provide the initial setup to organize the outcomes of challenges. The authoring tool will only record whether challenges have been (partly) completed by students. This status of the challenges can be consulted by authors and supervisors in the tool. The implementation for recording challenge results will be done in such a way that it can be easily extended by the learning analytics module.

The system will also provide access to the challenge results or statuses via web services. These web services will provide a starting point for the learning analytics module.

6.2.2 **Non-functional Requirements**

Besides the functional requirements we also define a number of non-functional requirements. These are quality constraints for the system. The international standard for software quality ISO/IEC 25010 ('ISO/IEC 9126', 2017), the successor of ISO 9126, contains a checklist for describing the non-functional requirements of a system. We have used this checklist for non-functional requirements elicitation, resulting in a set of non-functional requirement suitable for our system.

The list of non-functional requirements can be found in the next section. The first word in the description of each requirement refers to the software quality characteristic in the ISO/IEC 25010 standard. Subdivisions of a software characteristic in this standard have been indicated with an arrow (->).

Special attention has been given to the usability requirements. Good usability is very important for the authoring environment because is should be tailored towards casual users. Therefore we have explicitly formulated usability requirements. We have defined the usability requirements by using the following format. This format already includes the way to measure that the usability requirement is satisfied in the implemented system:

- Usability requirement: description of the requirement.
- User classes: the type of user who has that requirement. In our case a user can either be an author or a supervisor.
- Motivation: the reason why the usability requirement is important.
- Measuring concept: description of what will be measured to investigate whether the requirement is satisfied: user satisfaction, learnability, effectiveness, ease of use, quality of task performance
- Measuring method: description of how the measuring concept will be measured
- Worst level: minimal criterion to be met to pass the test for acceptance
- Planned level: the expected (normal) criterion for acceptance
- Best level: best possible level for the usability requirement

The reason for using this approach is to have a solid basis for the usability evaluation of the system. In chapter 8 we will describe the usability evaluation of the application. In this evaluation, we were able to revert to the measuring methods that we have defined here to check the system's usability.

6.2.3 List of Functional and Non-functional Requirements

Functional requirements

1. A supervisor will be able to create, edit, view and delete author accounts.
2. The user can add a student profile.
3. The student profile can contain information that is related to the context of school dropout.
4. The user can view, edit or delete a student profile.
5. The system will facilitate searching for a student profile.
6. The system will provide a web service for retrieving student profiles from the library.
7. The user can create different types of learning objects.
8. Learning objects must be suitable for use in a mobile web 2.0 environment.
9. The user can add a learning object.
10. The user can view, edit or delete a learning object.
11. The system will facilitate searching for a learning object.
12. The system will facilitate searching for learning objects that match a student's interest.
13. The system will provide a web service for retrieving learning objects and their metadata from the library.
14. The user can add a challenge.
15. The user can view, edit or delete a challenge.
16. The user can add challenges for a student in bulk.
17. The system will record the result of the challenge execution.
18. The system will provide access to the challenge results via a web service.
19. The user can consult the results of challenges.
20. The system will provide a web service for retrieving challenges and creating new challenges.

Non-functional, non-usability requirements

- Security->Confidentiality: a user management module with login functionality will be made available
- Reliability -> Recoverability: error messages should be logged in the system to help understand the cause of errors
- Maintainability->Changeability: the application will have a system documentation
- Maintainability->Changeability: the system will be built in a modular way, using only well-known technologies
- Portability->Adaptability: the user will only need a browser to use the application

Non-functional, usability requirements

Usability requirement 1	In each overview (students, learning objects and challenges) a user should be able to sort the results based on different criteria/properties.
User classes	Normal users
Motivation	Finding something in a long list may be difficult; sorting the list can help.
Measuring concept	quality of task performance
Measuring method	task scenarios Result: time required
Worst level	time required to find elements by criterion by sorting: 30 sec
Planned level	time required to find elements by criterion by sorting: 15 sec
Best case	time required to find elements by criterion by sorting: 10 sec

Usability requirement 2	In the learning objects overview, the user should be able to filter out results by applying a filter with the desired specification.
User classes	Normal users
Motivation	Finding something in a long list may be difficult; filtering the list can help.
Measuring concept	quality of task performance
Measuring method	task scenarios Result: time required
Worst level	time required to find elements by criteria with filter: 30 sec
Planned level	time required to find elements by criteria with filter: 15 sec
Best case	time required to find elements by criteria with filter: 10 sec

Usability requirement 3	In each overview (students, learning objects and challenges) a user should be able to search for the desired object by entering a search term. It should be possible to use incomplete info in the search function.
User classes	Normal users
Motivation	Finding something in a long list may be difficult; searching the list can help.
Measuring concept	Quality of task performance
Measuring method	task scenarios Result: time required
Worst level	time required to find elements by criteria: 30 sec
Planned level	time required to find elements by criteria: 15 sec
Best case	time required to find elements by criteria: 10 sec

Usability requirement 4	In each overview (students, learning objects and challenges) a default view with a selection of properties will be presented, with the option to view all details.
User classes	Normal users
Motivation	Presenting all information in the same view would make the user lose himself in too much details to find what he is looking for.
Measuring concept	Ease of use
Measuring method	Afterwards: questionnaire
Worst level	Good
Planned level	Very good
Best case	Excellent

Usability requirement 5	The user should be able to see an overview of the last actions that he performed .
User classes	Normal users
Motivation	If a user quickly wants to check whether or not he has already executed a task, he can check this in the overview.
Measuring concept	Ease of use, user satisfaction
Measuring method	task scenarios
Worst level	time required to find the last action: 30 sec.
Planned level	time required to find the last action: 15 sec.
Best case	time required to find the last action: < 10 sec.

Usability requirement 6	After finishing the ‘Start here!’ page from the user manual (approx. 20-25 min.), the user should be able to perform all basic tasks without extra training.
User classes	Normal users
Motivation	There is no time for extra training. Users can watch the video tutorials and/or follow the step-by-step walkthrougs (with screenshots). Learning to use the system is place and time independent.
Measuring concept	Learnability
Measuring method	task scenarios Number of times the user has to revisit the manual during task completion Afterwards: questionnaire
Worst level	Good
Planned level	Very good
Best case	Excellent

Usability requirement 7	Each overview (students, learning objects and challenges) should be reachable in one interaction with the system.
User classes	Normal user
Motivation	The options to see an overview should be directly available
Measuring concept	Effectiveness
Measuring method	Task scenario Ask the user in every state to return to the default state. Counting number of interactions with the system.
Worst level	Planned level
Planned level	Number of interactions with the system is 1.
Best case	Planned level

Usability requirement 8	It should be possible for an authenticated user to log off by one interaction with the system.
User classes	Normal user
Motivation	Logging off should always be possible when authenticated; logging off should be easy.
Measuring concept	effectiveness
Measuring method	Task scenario Result: number of times it isn’t possible to log off in 1 interaction
Worst level	Planned level
Planned level	number of times it isn’t possible to log off in 1 interaction is zero
Best case	Planned level

Usability requirement 9	Canceling should be available at each step of a process.
User classes	Normal user
Motivation	To avoid wrong input in the system, the user should be able to cancel at any time.
Measuring concept	quality of task performance
Measuring method	Task scenario Result: number of times the user wants to cancel but can't
Worst level	Planned level
Planned level	number of times the user wants to cancel but can't is zero
Best case	Planned level

Usability requirement 10	When the user is guided through a step-by-step process, in each step an overview with the progress (completed steps) and steps still to complete is presented.
User classes	Normal user
Motivation	Informing the user at any moment how much work still needs to be done and what is already completed will help him in performing the task.
Measuring concept	User satisfaction
Measuring method	Task scenario Result: number of times the user doesn't know how many steps he will still have to complete
Worst level	Planned level
Planned level	number of times the user doesn't know how many steps he will still have to complete is zero
Best case	Planned level

Usability requirement 11	When deleting an object (student profile, learning object or challenge) the system will always ask for confirmation before executing the delete.
User classes	Normal user
Motivation	The user doesn't want to delete an object by accident.
Measuring concept	quality of task performance
Measuring method	Task scenario Result: number of times an object is deleted without having to confirm
Worst level	Planned level
Planned level	number of times an object is deleted without having to confirm is zero
Best case	Planned level

Usability requirement 12	The overview of the challenges for a particular student and the overview of the challenges with a certain learning object should be represented in a visual manner, e.g. by a chart.
User classes	Normal user
Motivation	The user wants to see this information at a glance.
Measuring concept	Ease of use, effectiveness
Measuring method	Task scenario Result: user finds the number of challenges with a certain status in the visual representation without having to count. Afterwards: questionnaire
Worst level	time required to find the number of challenges with a status: 30 sec.
Planned level	time required to find the number of challenges with a status: 15 sec.
Best case	time required to find the number of challenges with a status: < 10 sec.

Usability requirement 13	In the views for adding or editing an object (student profile, learning object or challenge) over 75% of the fields are selection fields (checkbox, radiobutton, dropdown list, …).
User classes	Normal user
Motivation	The user wants to add or edit new student profiles, learning objects or challenges quickly, without having to do too much typing.
Measuring concept	Ease of use, effectiveness
Measuring method	Task scenario Result: The views for adding and editing contain over 75% selection fields.
Worst level	The views for adding and editing contain over 75% selection fields.
Planned level	The views for adding and editing contain over. 80% selection fields.
Best case	The views for adding and editing contain over 90% selection fields.

Usability requirement 14	A user should be able to add an appropriate standardized tag to a student profile's interest or learning object's topic in less than 1 minute.
User classes	Normal user
Motivation	The user wants to find and add tags quickly.
Measuring concept	Ease of use, effectiveness
Measuring method	Task scenario Result: The tag is found and added within a minute.
Worst level	The tag is found and added within two minutes.
Planned level	The tag is found and added within a minute.
Best case	The tag is found and added within 30 sec..

6.3 Use Cases

6.3.1 Use Case Diagram

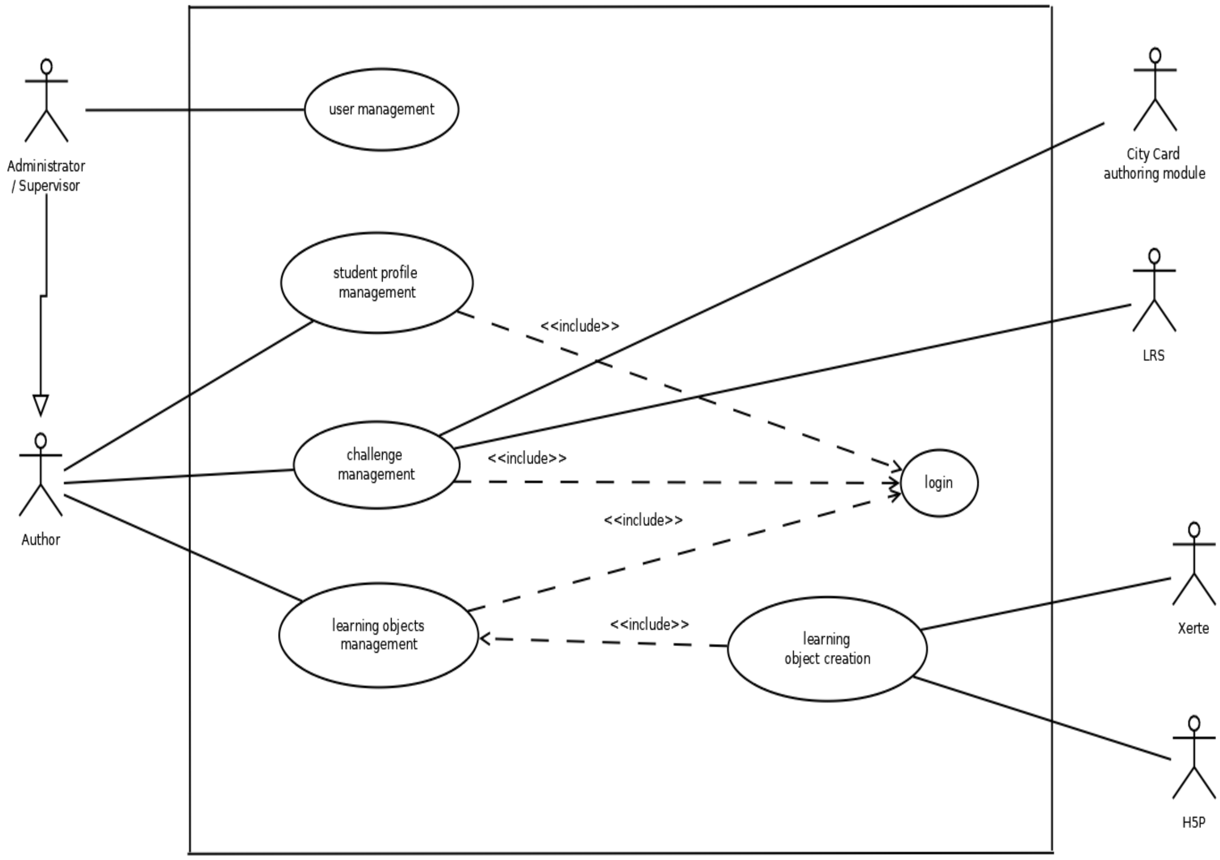


Figure 9 Use Case Diagram For the Authoring Environment

The use case diagram captures the functional requirements of the authoring tool at the highest level. The complete system is divided into five main use cases: user management, student profile management, challenge management and learning objects management. In the next section, for each use case basic and alternative scenarios are given. The login part of the system is not further described.

6.3.2 Use Case Scenarios

Use case 1: user management

Primary actor: administrator/supervisor
Goal: User management = manage accounts for authors' access to the system (create, view, edit, delete)
Stakeholders and interests:

- Administrator/supervisor: wants to create, view, edit or delete an author account
- Authors: wants to be able to use the system

Preconditions: The administrator is identified and authenticated
Postconditions: Changes to the author accounts have been saved

Main Success Scenario (Basic Flow):

1 The administrator starts the user management.
2 The system presents an overview of existing users (= authors).
3 The administrator selects the author he wants to view the details from.
4 The system presents the detailed information of the selected author (first name, last name, school, password, rights).
5 The administrator confirms.

Extensions (Alternative Flow):

1 -
2 Requesting the existing author fails.
   The system responds with an error message.
3a The administrator wants to create a new author account.
   1 The system requests for the information to create a new author account (first name, last name, username, password and rights).
   2 The administrator completes at least the required fields and confirms.
   Alternative scenario:
      1 The administrator fails to complete all required fields.
      2 The system indicates which fields still need to be completed.
      3 Back to alternative scenario 3a.1
   3 The system confirms the successful creation of the author.
   4 Back to main success scenario step 2.
3b The administrator wants to edit an existing author account.
   1 The system presents the available information of the author account (first name, last name, username, password and rights).
   2 The administrator edits the author account information, making sure the minimal required fields are completed.
   Alternative scenario:
      1 The administrator doesn't complete all required fields.
      2 The system indicates which fields still need to be completed.
      3 Back to alternative scenario 3b.1
   3 The system confirms the changes have been saved.
   4 Back to main success scenario step 2.

3c The administrator wants to deactivate an author.

- 1 The system asks for confirmation.
- 2 The administrator confirms.

  Alternative scenario:

  - 1 The administrator cancels.
  - 2 Back to main success scenario step 2.
- 3 The system confirms the deactivation of the author.

3d The administrator wants to activate an author.

- 1 The system asks for confirmation.
- 2 The administrator confirms.

  Alternative scenario:

  - 1 The administrator cancels.
  - 2 Back to main success scenario step 2.
- 3 The system confirms the activation of the author.

Use case 2: student profile management

Primary actor: author
Goal: Student profile management = manage the student profiles in the system (create, view, edit, delete)
Stakeholders and interests:

- Author: wants to create, view, edit or delete student profiles

Preconditions: The author is identified and authenticated
Postconditions: Changes to the student profiles have been saved

Main Success Scenario (Basic Flow):

1 The author starts the student profile management.
2 The system presents an overview of existing student profiles.
3 The author selects the user profile he wants to view the details from.
4 The system presents the detailed information of the selected student profile.
5 The author confirms.

Extensions (Alternative Flow):

1 -
2 Requesting the existing student profile fails.
  The system responds with an error message.
3a The author wants to create a new student profile.
  1 The system requests the information to create a new student profile.
  2 The author completes at least the required fields (first name, last name, gender and LRS credentials) and confirms.
    Alternative scenario:
    1 The author fails to complete all required fields.
    2 The system indicates which fields still need to be completed.
    3 Back to alternative scenario 3a.1
  3 The system confirms the successful creation of the student profile.
  4 Back to main success scenario step 2.
3b The author wants to edit an existing student profile.
  1 The system presents the available information of the student profile.
  2 The author edits the student profile information, making sure the minimal required fields (first name, last name, gender and LRS credentials) are completed.
    Alternative scenario:
    1 The author doesn’t complete all required fields.
    2 The system indicates which fields still need to be completed.
    3 Back to alternative scenario 3b.1
  3 The system confirms the changes have been saved.
  4 Back to main success scenario step 2.
3c The author wants to delete a student profile.
  1 The system asks for confirmation.
  2 The author confirms.

Alternative scenario:

1 The author cancels.

2 Back to main success scenario step 2.

3 The system confirms that the student profile has been deleted.

Use case 3: learning objects management

Primary actor: Author
Goal: Learning objects management = manage the learning objects in the system (create, view, edit, delete)
Stakeholders and interests:

- Author: wants to create, view, edit or delete learning objects

Preconditions: The author is identified and authenticated
Postconditions: Changes to the learning objects have been saved

Main Success Scenario (Basic Flow):

1 The author starts the learning objects management.
2 The system presents an overview of existing learning objects.
3 The author selects the learning object he wants to view the details from.
4 The system presents the detailed information of the selected learning object.
5 The author confirms.

Extensions (Alternative Flow):

1 -
2 Requesting the existing learning object fails.
  The system responds with an error message.
3a The author wants to add a new learning object.
  1 The system requests for the information to add a new learning object.
  2 The author completes at least the required fields (title, select Xerte .zip for upload or the embed code for H5P, education level, education year, difficulty, paedagogic goal, interactivity level, language and language level) and confirms.
   Alternative scenario:
    1 The author fails to complete all required fields.
    2 The system indicates which fields still need to be completed.
    3 Back to alternative scenario 3a.1
  3 The system confirms the learning object has been added or presents an error message that something went wrong.
  4 Back to main success scenario step 2.
3b The author wants to edit an existing learning object.
  1 The system presents the available information of the learning object.
  2 The author edits the learning object information, making sure the minimal required fields (title, select Xerte .zip for upload or the embed code for H5P, education level, education year, difficulty, paedagogic goal, interactivity level, language and language level) are completed.
   Alternative scenario:
    1 The author doesn't complete all required fields.
    2 The system indicates which fields still need to be completed.
    3 Back to alternative scenario 3b.1
  3 The system confirms the changes have been saved or presents an error message

that something went wrong.

4 Back to main success scenario step 2.

3c The author wants to delete a learning object.

1 The system asks for confirmation.

2 The author confirms.

Alternative scenario:

1 The author cancels.

2 Back to main success scenario step 2.

3 The system confirms that the learning object has been deleted.

Use case 4: challenge management

Primary actor: author or admin/supervisor, from here referred to as *actor*
Goal: Challenge management = manage the challenges in the system (create, view, delete)
Stakeholders and interests:

- Supervisor: wants to create, view or delete learning objects of an author
- Author: wants to create, view or delete his/her learning objects

Preconditions: The actor is identified and authenticated
Postconditions: Changes to the learning objects have been saved

Main Success Scenario (Basic Flow):

1 The actor starts the challenge management.
2 The system presents an overview of existing challenges.
3 The actor selects the challenge he wants to view the details from.
4 The system presents the detailed information of the selected challenge.
5 The actor confirms.

Extensions (Alternative Flow):

1 -
2 Requesting the overview of challenges fails.
  The system responds with an error message.
3b The actor wants to add a new challenge.
  1 The actor selects the author of the challenge.
  2 The system presents the overview of all students.
  3 The supervisor selects the student for whom the challenge will be created.
  4 The system presents all learning objects that are available to use in challenges.
  5 The supervisor selects the learning objects he wants to use in the challenges.
  6 The system presents an overview of the challenges that will be created.
  7 The supervisor confirms.
  8 The system creates a new challenge for each selected learning object for the student.
  9 The system presents a message upon successful creation or an errormessage if something went wrong.
  10 Back to main success scenario step 2.
3c The actor wants to delete a challenge
  1 The system asks for confirmation.
  2 The actor confirms.
  Alternative scenario:
    1 The actor cancels.
    2 Back to main success scenario step 2.
  3 The system confirms that the challenge has been deleted.
  4 Back to main success scenario step 2.
3d The actor wants to view the challenges' results of a student
  1 The actor starts the challenge overview by student.
  2 The system presents a list of all available students.

3 The actor selects the student.
4 The system presents an overview of the selected student's challenges

3e The actor wants to view the challenge results of a learning object

1 The actor starts the challenge overview by learning object.
2 The system presents a list of all available learning objects.
3 The actor selects the learning object.
4 The system presents an overview of the challenges with the selected learning object.

6.4 Summary

We started this chapter with a description of the context of the authoring tool. We presented the TICKLE project as the general context and explained how the authoring tool will relate to the other parts of TICKLE's system architecture.

In the rest this chapter we have set out the requirements that the authoring tool should meet. We have defined the functional requirements and non-functional requirements. For the non-functional requirements we have put focus on the usability requirements, since the tool shall be used by *casual users*. All usability requirements contain a description, a motivation, measuring methods and acceptance criteria that can be used for a usability evaluation of the tool.

Use case scenarios have been defined for the four main parts of the tool: user management, student profile management, learning objects management and challenge management. All use case scenarios describe the main success scenario (basic workflow) and the possible alternative flows.

7

Design and Implementation

7.1 Introduction

Based on the preliminary background work in chapters 2, 3, 4 and 5 and the requirements elicitation in the previous chapter, we can now elaborate in this chapter on the contributions made in this thesis. In this chapter we will describe the authoring tool we have build in order to fulfil the requirements we have defined. We will present the used technologies and the design of the tool. Finally, we will discuss the proposed solution to a number of noteworthy issues we encountered during implementation.

7.2 Technology Choices

As shown in the system architecture, the authoring tool we are developing is loosely coupled with the rest of the TICKLE modules. Communication with the City Card Authoring environment will happen via web services. Therefore, in choosing the technology for the authoring environment we had a certain amount of freedom.

To avoid that the user has to install the application on his computer, we opt for a web application. In this way the application will be platform (Windows, Linux, Mac) and device (PC, laptop, tablet) independent. Interaction with the system is through a web browser.

For the development of the web application it was important to choose the right technology. We prefered to use a framework, because frameworks tend to contain standardized solutions for well-known problems. After investigating different web application frameworks, we decided to go for CodeIgniter[2]. Since version 4 is still under development, we have used CodeIgniter3 in this project. These were the reasons for choosing this framework:

- Effort to learn: CodeIgniter has a short learning time. From following some basic tutorials it became clear that the workflow CodeIgniter imposes is quite straightforward.
- Documentation: The CodeIgniter documentation is organized in a very logically way and provides tutorials for all parts of the framework
- Popularity: The framework has been around for over a decade and has become one of the leading PHP frameworks. This is important because nearly every question that came up during development has already been answered on websites like stackoverflow.com.
- MVC opinionated: CodeIgniter encourages the Model-View-Controller (MVC) design. Based on experience with the MVC pattern in programming projects, we considered this as a big plus.
- Security: This factor has become more and more important. CodeIgniter provides built-in protection against CSRF and XSS attacks. If the guidelines are followed, CodeIgniter also provides protection against SQL injection.

For the database we have chosen to use SQLite[3]. SQLite is a single-file database that requires no extra server thread. Since it behaves as a file, it can be used across all platforms. The source code is publicly available and it has the reputation of being very well tested.

For the front-end of the application we opted for the Twitter Bootstrap framework in combination with CSS3 and JQuery. Twitter Bootstrap is a front-end library to develop websites and web applications. It supports all major browsers (even legacy browsers back to Internet Explorer 8) and supports responsive web design, thus ensuring the user interface will look the same on all devices and all browsers. Twitter Bootstrap 4 was released in January 2018.

7.3 Design

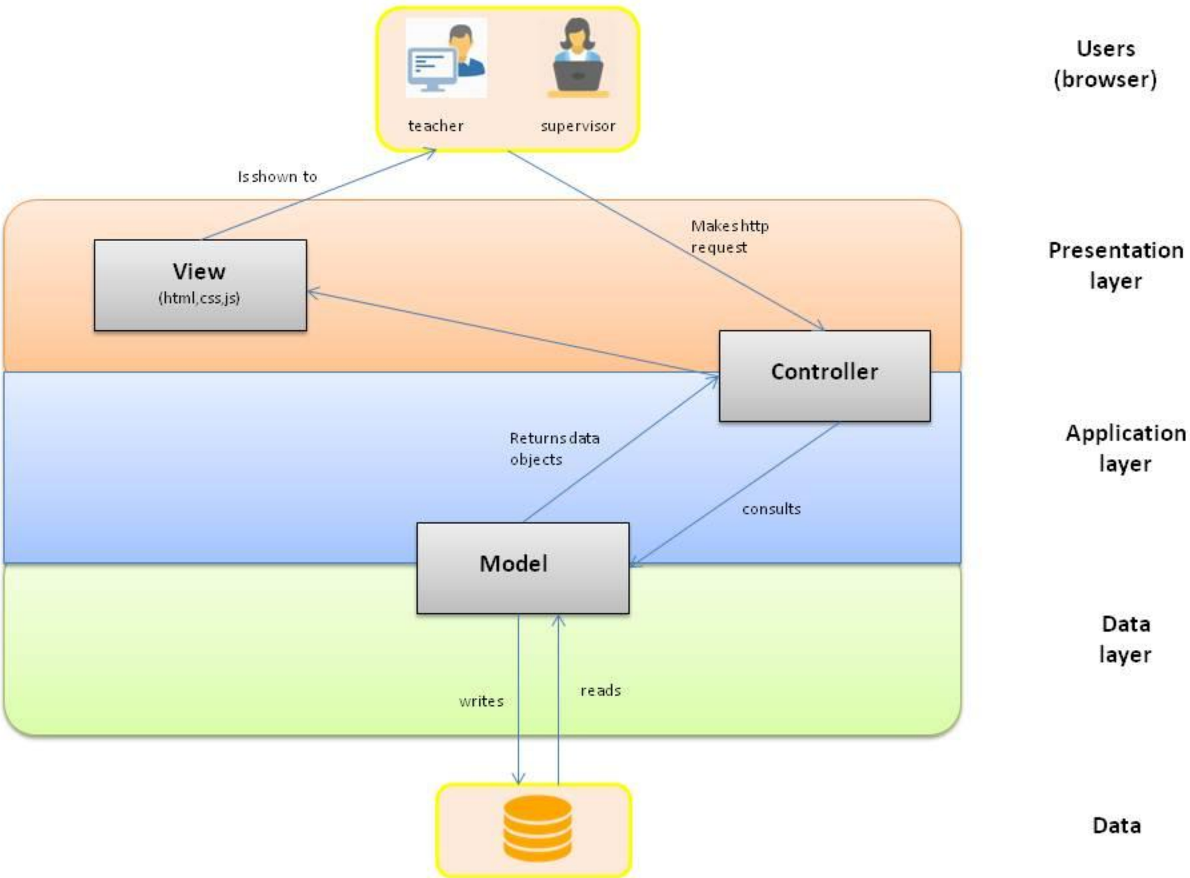
7.3.1 MVC Architecture in CodeIgniter 3

CodeIgniter is opinionated, meaning that it enforces the developer to achieve certain functionality in a predefined way. When using CodeIgniter as development framework, the programmer (by default) is enforced to use the MVC pattern, thus ensuring there will be a clean separation between the data objects in the Model and the representation

[2] https://www.codeigniter.com/, accessed on 03/04/2018
[3] https://www.sqlite.org, accessed on 03/04/2018

Figure 10 MVC Architecture in CodeIgniter 3

to the user in the View. The following diagram describes how the Model-View-Controller is embedded in CodeIgniter.



The typical workflow goes as follows:

1. User sends a http request to the system
2. The controller determines the request type and looks up the corresponding action.
3. That controller action consults the model to get the data objects necessary to handle the action.
4. The model accesses the database and returns the data objects that were requested.
5. The controller manipulates the data objects in order to contain all information to fill up the view.
6. The corresponding view is called from the controller action and the data objects are passed.
7. The view is rendered and shown to the user.

7.3.2 Database Model

A simplified version of the data model is shown in Figure 11. Some tables have been left out to keep the model readable.

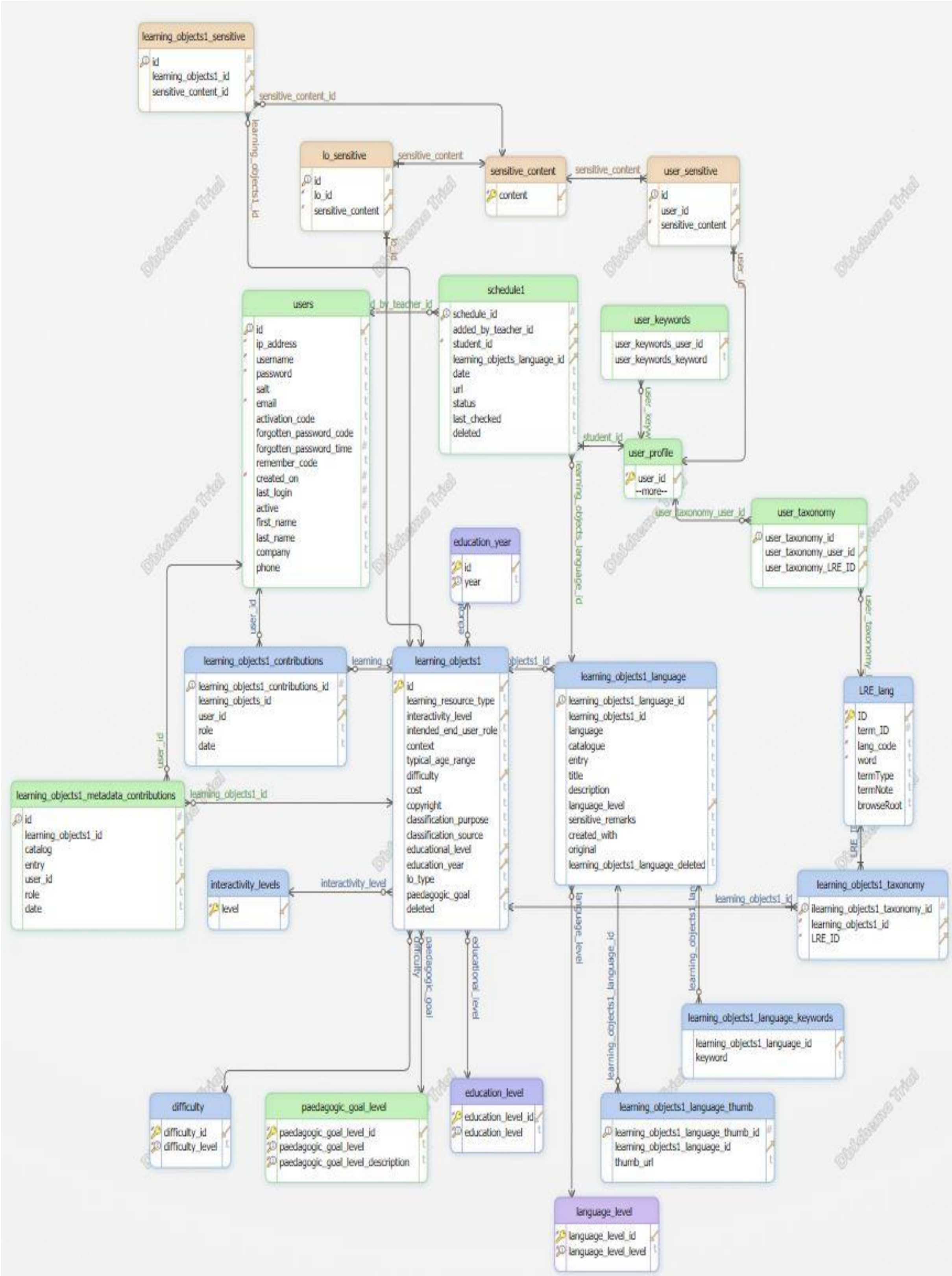


Figure 11 Database Model

In the remainder of this paragraph we will explain how the tables contain the necessary information for student profiles, learning objects, and challenges.

Learning objects

The main table for the learning objects is *learning_objects1*. This table's columns contain the core values of a learning object, i.e. the characteristics that are not language dependent. Because of the ILOX-LOM standard (discussed in section 5.6) that we have chosen for representing a learning object's metadata, the need for a supplementary table *learning_objects1_language* arises. The ILOX-LOM standard defines that a learning object can have translations in different languages. Hence, we need to keep the language specific metadata of a learning object separate from the language independent properties.

As can be seen from the database model in Figure 11 the columns of the table learning_objects1 will indeed not change in translations of the learning object: *difficulty, pedagogic_goal, educational_level, education_year, ...* For standardization purposes, their values will be a foreign key, referring to the ID of a table holding all possible values for that field.

The columns in the table *learning_objects1_language* hold the language specific properties of a learning object: *language, language_level, title, description… Catalogue* and *entry* combined (with a slash) are used to compose the unique URL where the learning object is physically stored on the server.

The only tags that can be used to describe the learning object are the ones from the LRE Thesaurus. The LRE Thesaurus itself can be found in the table LRE_lang. The table *learning_objects1_taxonomy* is an intermediate table, used to contain the different tags from the thesaurus for each learning object. A user will also have the possibility to add his own, more specific tags to describe the learning object. For this, the table *learning_objects1_language_keywords* is used.

To abide by the ILOX-LOM standard we also need to register the contributions of authors to a learning object. A learning object may, for instance, receive an update from its original author, or the learning object could be translated into a different language by another author. These kinds of contributions will be kept in the table *learning_objects1_contributions*. The content that some students might find intrusive is held in the table *learning_objects1_sensitive*.

Student profiles

The central table for the student profile management is the table *user_profile*. Because the visual representation of this table in the diagram would take up too much space, the columns have been omitted, except for the primary key (user_id). The table contains seventy-five columns that can be used to complete the student profile. The seventy-five columns match exactly with the list of all possible elements for a student profile related to school dropout composed in section 4.10.

The tables *user_keywords* and *user_taxonomy* hold the information on the student's interests. The interests in the table *user_taxonomy* are described according to the taxonomy of the ILOX-LOM standard. As described there, we will use this standard to model student interests as well as learning objects topics. This will allow us to find learning objects that suit the student's interest. The only tags that can be used to model user interest are the tags from the LRE Thesaurus, which is held in the table *LRE_lang*. This table contains the available main tags in different languages, hence the second part of the table name *LRE_lang*.

Besides the tags, there will also be the possibility to use keywords to describe a student's interest. In contrast to the tags from the ILOX taxonomy, the keywords can be freely chosen by the user. For this purpose the table *user_keywords* is used. The content that a student might be offended by is held in the table *user_sensitive*.

Challenges

The *challenges* table is where everything comes together. Since a challenge is made by an author for a student and contains a learning object, these three ids are necessary here. Note that for a challenge we need the id from the table *learning_objects1_language*.

The challenges table also contains the URL to the actual challenge to be completed by the student. This URL can then be used in the City Card Authoring Environment to attach the challenge to a card and location. Finally, we also allow to record the status of the challenge.

7.3.3 Design diagram

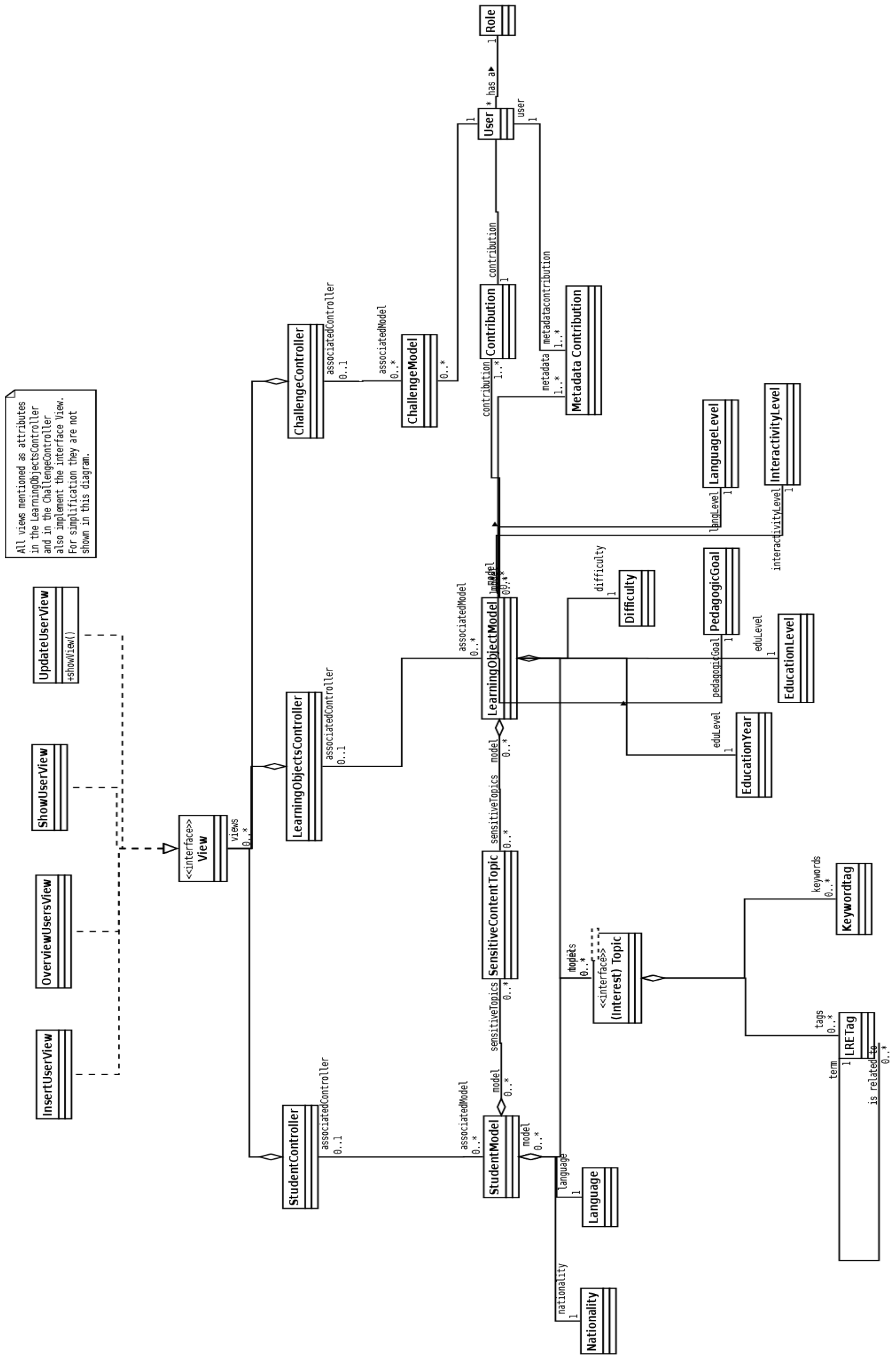


Figure 12 Simplified UML Class Diagram

Figure 12 gives the general overiew of the system design by means of a simplified UML class diagram. For readability reasons, only the main classes of the application are shown. Attributes and methods have been omitted. Note that a number of the supporting classes such as Services and Authorization have been left out. The class Services bundles all services that are needed in the controllers. The Authorization class handles all authorization in the application. Also the RESTControllers are omitted from the diagram. We have made three different RESTControllers: one for the student, one for the learning object and one for the challenge. Each RESTController can define its own methods for the typical REST methods: GET, POST, PUT and DELETE.

As discussed earlier, the application uses the Model-View-Controller design. We will briefly discuss the main parts of the design in the remainder of this section. We will start with zooming in on how the three main parts of the application have been designed: the student, the learning object, and the challenge. Afterwards we will explain the Interest/topic connection between the *Student* and the *LearningObject*.

The student

The StudentModel is the part of the application that handles the data of the student profile. A StudentModel object contains all data part of the student profile. These are all the data that have been listed in section 4.10.

The StudentModel is used in the StudentController. As discussed in the general overview on MVC in CodeIgniter (section 7.3.1), the StudentController will use the StudentModel to update the different Views. The StudentController will accept all requests made by the user of the system. Each of those requests is handled in a public function in the StudentController: add, deleteUser, edit, getUser, insert, overview, showUser, update.

When a request is made by the user, the StudentController will consult the StudentModel to gather the data needed. The StudentController will update the associated view with these data.

Just as both other controllers that will be discussed on the next pages, the StudentController uses the *Ion_auth* class. This class handles all authorization in the application. As it properties it keeps the current user (from class User). It has some methods to check if a user has the correct rights to execute the task he requests and methods to login and logout.

The Learning Object

The LearningObjectModel is the part of the application that handles the data of the learning object. A LearningObjectModel object contains all data that are part of the learning object as its properties. The model has methods to communicate with the database: insert and update.

For handling the contributions to a learning object we have made two separate classes. The Contribution class handles the contributions to the content of the learning object itself. Typical things that fall into this category are making a translation of an existing learning object or updating the content of the learning object.

The MetadataContribution class models the contributions to the metadata of a learning object, i.e. it keeps track of which users have updated the metadata at what time. We have modeled the contributions to the metadata in a separate class, since we have decided to use the standard ILOX and an IEEE LOM selection as standard for representing the metadata. In this standard, a clear distinction has been made between both, as has been discussed in section 5.6.5.

All contributions, either to the learning object or to its metadata, are done by users. In our case we will have two types of users: teachers and supervisors. To model this we have made a class *User* with all common features a user of the system should have, but we also made a class role. For now there will be only two roles (teacher or supervisor), but by modeling it in this way, it will be very easy to extend the application with new types of users by adding more roles.

The challenge

In the challenges everything comes together. Since a challenge contains a learning object and is made by a teacher for a student, the ChallengeModel uses all other available models by aggregation: LearningObjectModel, StudentModel and User. Note that the connections to the LearningObjectModel and StudentModel have been omitted for readability of the diagram. The functions provided in this model are insert and delete.

The ChallengeModel is used in the ChallengeController. This controller *catches* all requests related to challenges that are made by users. For each of these requests a method is provided in the controller.

The interest/topic and sensitive content mapping

One of the key challenges in this thesis was to find an acceptable solution for the mapping of the student's interest to the learning object's topics. We have discussed the proposed solution in section 5.7. We decided to use the same tagging system for the student interests and the learning object topics. Besides the tags we also have the keywords, freely specified by the user, to determine user interest and learning object topic. This enables us to establish a mapping between both, so we can find learning objects that are suitable to the student's interest.

We have used the tags from the LRE for schools. These have been modeled in the design in the class *LRETag*. Each LRETag is related to one or more other LRETags. This relation will help users to reveal related tags that might be more suitable for the interest or topic. Since the keywords can be chosen freely by the user, the class *KeywordTag* does not contain any relations between keywords.

Both classes are used to model a student's interest or a learning object's topic. An Interest/topic can contain multiple tags and keywords.

In the same way as the Interest/topic we have also modeled the possible sensitive content. For this purpose the class SensitiveContentTopic has been created. Just as with Interest/topic, a StudentModel and LearningObjectModel may contain multiple sensitive content topics.

7.4 Implementation

Requirement 17 states that the system should record the result of learning activities. As discussed in section 2.2, we decided to use xAPI for tracking student results. Therefore, the challenges should be made xAPI-ready, i.e. code should be added upon challenge creation in order to keep track of the student's results when the challenge is executed by the student. In this section we will discuss how this is handled for Xerte and H5P learning objects. Afterwards, we will present our RESTful implementation of the webservices that the tool provides. Next, we will discuss how the Tag Explorer and Matcher have been implemented. Finally, we will present our solutions for facilitating search and challenge results visualization.

7.4.1 Making Xerte Learning Objects xAPI-ready

As we decided to use xAPI for storing learning activities and Xerte for creating learning objects, we had to find a way to convert the learning object provided by Xerte into the xAPI format, as Xerte does not support xAPI yet.

Once an author has finished creating a learning object, it needs to be exported in order to be usable. Since we planned to target a mobile environment for our learning objects, we will always choose for the HTML5 export option instead of Flash. Xerte has four HTML5 export possibilities for learning objects:

- A deployment .zip, without SCORM tracking
- An archive .zip
- A SCORM package (1.2 or 2004) with tracking possibilities
- An offline .zip

The first two options are mainly there for portability reasons, e.g. exchanging learning objects between different Xerte environments. A user can make objects in his Xerte Online Toolkits environment, export them with one of both first options and mail them to his friend. The friend can then import the project, make changes if needed and export in the right format.

The third option is to export the project as a SCORM package. This SCORM package must then be uploaded in an LMS, where it can be launched to the students. Note that launching a SCORM package without an LMS is not possible. All data tracking of the students' activities happens within the LMS, where it can be accessed by the teacher.

The fourth option is to export a Xerte learning object as an offline .zip. When extracted, this package contains a file index.html that can be launched in a browser.

For our project we will use this fourth option, as exporting a learning object as a SCORM package (3th option) would require an LMS to launch it. Our aim is not to store the recorded data in an LMS with SCORM, but to store them in an LRS with TinCan/xAPI, as has been discussed in section 2.2. For using TinCap/xAPI there is no need to pass through SCORM first.

Therefore, we will use the offline .zip file and add JavaScript code to send the learning experiences to the LRS. Because the complete learning analytics will be the subject of a separate module, we will limit ourselves here to simply recording the status of a learning activity: *open, started, completed*. The status *open* means the learning activity is available for the student, but wasn't started yet. When the student clicks the link to start the learning activity, the status will be changed to *started*. The status will remain *started*, until the student has successfully completed the learning activity. Then the status will be changed to *completed*.

When a Xerte learning object is added to the library, the user has to upload the exported .zip-file. In the background, this file will be extracted in the *learning_objects_base* folder. When an actual challenge needs to be created, an instance of the base object is created in the *learning_objects_students* folder. When this version for the student is being created a copy of the base object is made and the code for sending the xAPI statements to track student results is added. To add xAPI tracking to the Xerte learning objects, these are the 4 steps the application will execute in the background when creating a challenge for a student.

1. Add the files *Tincan.js* en *xapiconfig.js* to the challenge folder

Tincan.js is an API provided by Rustici and ADL for communication with a LRS. xapiconfig.js is a configuration file we have written ourselves, containing all LRS configuration needed for our project. Upon creation of a challenge, an instance of the xapiconfig.js is added to the challenge folder, containing the LRS endpoint and the LRS login credentials: username, password and e-mail (mbox). This information is taken from the student profile of the student for which the challenge has been created. In the *init* function, a new Tincan javascript object from the API is created and initialized with the LRS credentials.

The file *xapiconfig.js* also contains a function *buildStatement* to build up a statement to be saved in the LRS, using the rules of the controlled vocabulary. This function builds xAPI statements following the actor/verb/object pattern. It also adds the result to the statement. The function *sentXAPIstmt* in *xapiconfig.js* then takes this created statement as input and uses the *Tincan.js* API to send it to the Learning Record Store.

2. Add code to index.html

Each Xerte learning object has an index.hml file as starting point. In the index.html of the challenge, we will add both JavaScript files described in step 1, so they can be accessed to send statements to the LRS when the challenge is started and completed.

3. Replace the xttracking_noop.js file

Each Xerte learning object contains a file xttracking_noop.js. This file works as an interface for some important functions that can be used for tracking the student interaction with the learning object. We have made an xAPI-ready version of this xttracking_noop.js file. Our version adds some additional code to some of the key tracking functions in xttracking_noop.js: XTInitialize, XTEnterPage, XTExitPage and XTSetPageScore. Each time one of these functions is called when the student is executing the learning activity, a check is executed to determine if some statement needs to be send to the LRS. Besides these Xerte functions we also mention our own function *sentIfAllCompleted*. Example code we dynamically add to the functions mentioned above can be found in Appendix C.

4. Solve the *special cases*

Although the *xttracking_noop.js* file is used for tracking the student during learning activities, we have noticed that the tracking file does not handle all possible scenarios. The functions XTEnterPage and XTExitPage are not used in some types of interactive exercises or games: *hangman, memory, gap fill, categories* and *text match*. For those types of activities, this extra step 4 is required.

Every Xerte learning object has one main JavaScript file that contains all events and the handling of those events. For each type of exercise, this main file is not dependent on the rest of the learning object's properties. The main file of a categories exercise for instance, will always be the same *categories.js* file. For the five types of exercises we mentioned where xttracking_noop.js doesn't handle all tracking, we have hacked into the code of the main file, to send the LRS statements there, if applicable.

So in this final step, the application will check if the learning object is one of those five types of exercises (hangman, memory, gap fill, categories and text match). If it is, it will replace its main file with the file containing code for sending the xAPI statements to the LRS.

7.4.2 Making H5P Learning Objects xAPI-ready

Learning objects that are created with H5P need to be added in a different way than Xerte learning objects. H5P learning objects are not uploaded to a folder in the application. For adding an H5P learning object to the learning object's library, we add an empty index.html file and copy the embed code to an iframe in that index.html file.

Whenever an actual challenge for a user is made with this learning object, some extra JavaScript code will be added to this index.html file. Upon creation, we will copy code to send xAPI statements to the LRS if the learning activity is completed.

When a learning object is opened, an xAPI statement will be send to the LRS that the student has started the challenge (i.e. launched). The added JavaScript code will then catch clicks in the iframe where the learning object is situated and look for the score summary on the last slide of a H5P course presentation. If the score is sufficient to pass the learning activity an xAPI statement will be sent to the LRS to change the status to *completed*.

Note that this approach only works for H5P course presentations. However, as explained in the documentation, this does not impose much limitations to the different sorts of interactivity exercises available. Indeed, an H5P course presentation can contain nearly all available types of interactivity on the different slides.

Finally we should also notice that with this approach it is necessary to host the H5P authoring tool on the same server as the authoring tool we are developing. Trying to make JavaScript calls from inside the iframe, would result in a cross origin request error.

The approach presented for Xerte and H5P is our proposed solution for requirement 17:

- *The system will record the result of the challenge execution.*

7.4.3 **REST APIs**

As mentioned above in the list of requirements (section 6.2.3) the system also needs to provide web services for retrieving student profiles, learning objects, and challenges. The web services also need to provide functionality to create a new challenge. These are requirements 6, 13, 18 and 20 from the requirements list.

For the web service implementation we have decided to use REST instead of SOAP. REST does not require as much overhead and has a better performance, because it easier to parse and enables caching. However, the main factor for preferring REST is that it permits many data formats, whereas SOAP is limited to XML. The developer of the City Card authoring module, who will mostly be consuming these services, prefers the JSON format to obtain the data, so REST was the obvious choice.

For implementing the REST controller, we used the CodeIgniter *REST_controller* class from Phil Sturgeon and Chris Kacerguis[4]. The class functions as a library that can be added to the application and contains some interfaces to be implemented, with which a REST service can be made.

These web services will be used by the city card authoring module, as illustrated in the TICKLE system architecture in section 6.1.4. Besides the functionality mentioned in the requirements, we agreed with the developer of the city card authoring module to add some more specific functions in the web service. An overview of the 19 types of available requests to the REST API, together with a typical JSON response can be found in Appendix D.

The approach presented in this paragraph is our proposed solution for requirements 6, 13, 18 and 20:

[4] https://github.com/chriskacerguis/codeigniter-restserver, accessed on 04/04/2018

- *The system will provide a web service for retrieving student profiles from the library.*
- *The system will provide a web service for retrieving learning objects and their metadata from the library.*
    - *The system will provide access to the challenge results via a web service.*
    - *The system will provide a web service for retrieving challenges and creating new challenges.*

7.4.4 Tag Explorer and Matcher

One of the challenges in this thesis was establishing the link between user interests and learning objects. In section 5.7 we have give a detailed explanation on why we will use the same controlled vocabulary for modeling user interests and learning object topics. We developed a user interface, called *Tag Explorer* to facilitate the search for the correct description tags. Remember that facilitating this process was explicit part of the requirements

To facilitate the choice of the right tags from the controlled vocabulary for the teacher, a user-friendly interface (*Tag Explorer*) was developed. A teacher can start searching for the most suitable tags by selecting a main topic. In the *Tag Explorer*, all related subjects from the LO Thesaurus will be shown as possible candidates for tags. If a user wants to find topics related to any of the possible tags, he can just click the magnifying glass next to the subject. For user convenience the list of the last searched tags is kept. To select a tag for a learning object, the user can simply drag and drop the tag from the tag explorer to the *selected tags* field. In Figure 13 an example is given; the only selected tag is 'Trigoniometry'. Removing a tag from the selection also happens through drag-and-drop the tags to the trash.

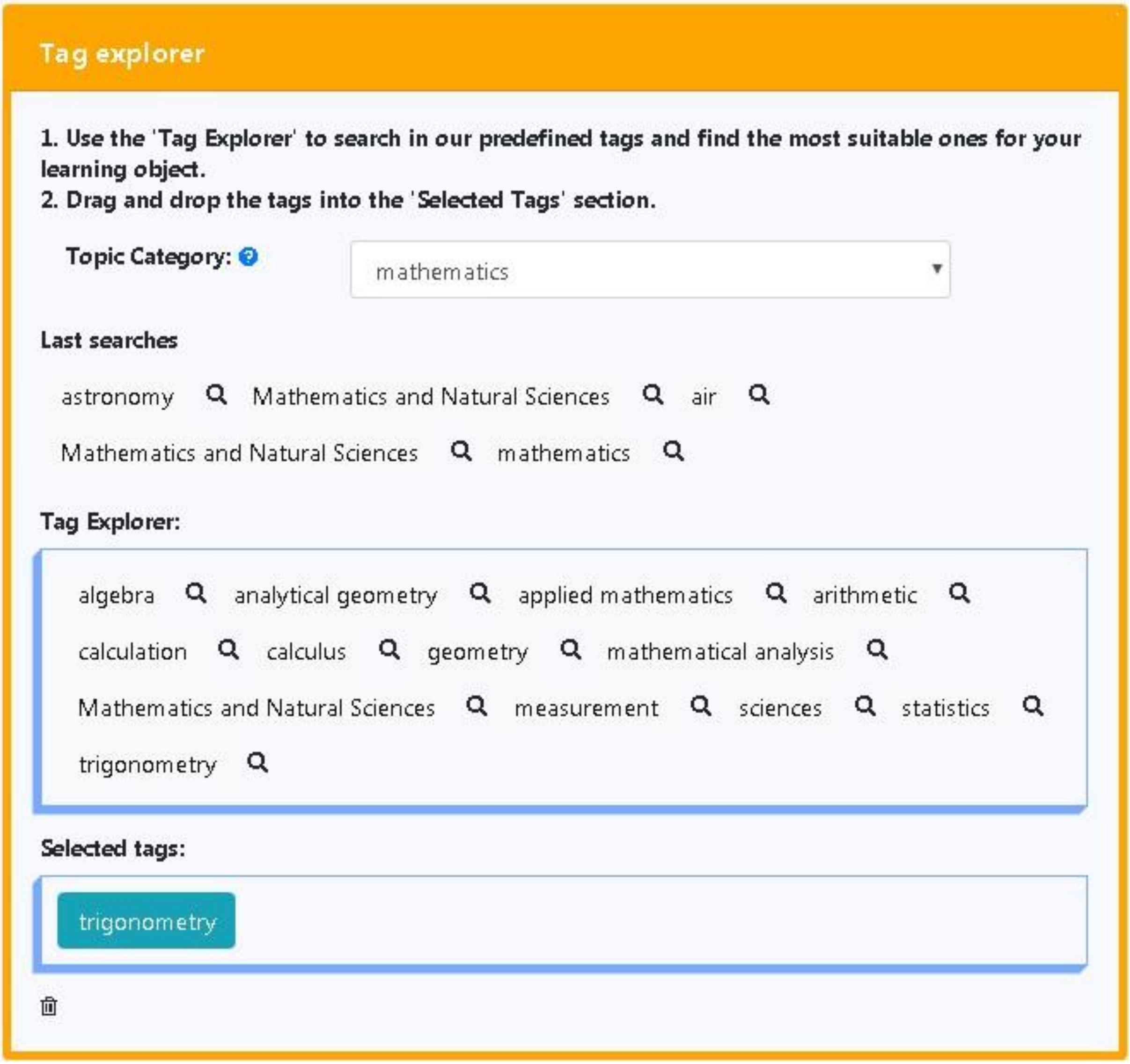


Figure 13 Tag explorer: GUI For Adding Tags To a LO

As explained in section 5.7 we will use the same controlled vocabulary from the LRE Thesaurus to model user interests. This implicates that we will use the Tag Explorer also in the GUI for adding and editing a student profile.

Using the same tags for student interests and learning object topics, allows us to find the most suitable learning objects in the database according to the student interest. This matching takes place in the Matcher, where the teacher will select a student and an ordered list of learning objects will be shown, ordered by relevance for the student. The algorithm to do this has been already explained in detail in section 5.7.

The approach presented in this paragraph is our proposed solution for requirement 12: *The system will facilitate searching for learning objects that match the student interests*. We have developed the Tag Explorer in order to meet usability requirement 14: *A user should be able to add an appropriate standardized tag to a student profile's interest or learning object's topic in less than 1 minute.*

7.4.5 Facilitating learning objects search

When the learning objects library becomes very large, it will become harder for users to find the learning object they are looking for. The application provides three ways to facilitate searching for the desired learning object: filters, search field, and sorting.

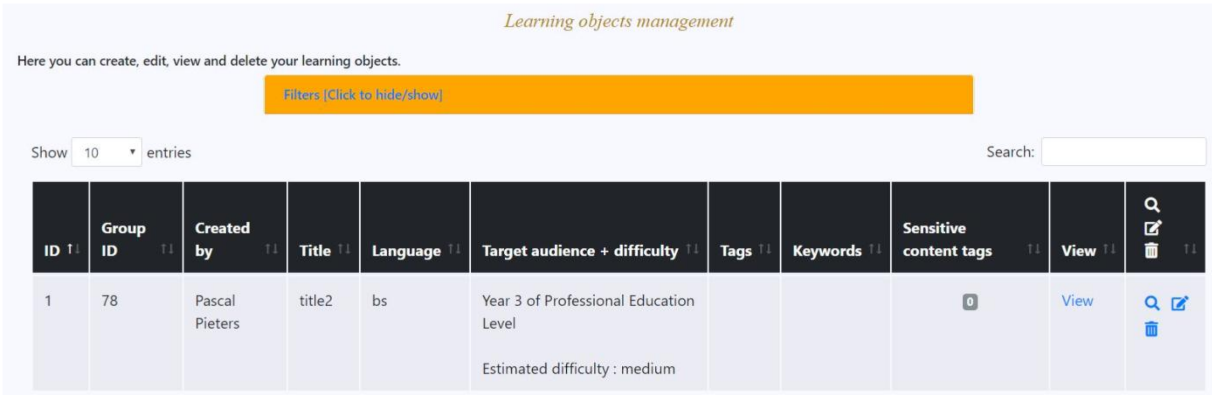


Figure 14 Facilitating Learning Objects Search

- Sorting

The learning objects library presents the metadata of each learning object as a single row with values in different columns (see Figure 14). By pressing the column headers the rows will be sorted in ascending or descending order. Sorting alphabetically by title or language could provide a way to quickly find what one is looking for.

- Search field

Right above the table with the learning objects metadata a search field can be used to search in the learning objects for a certain keyword. The search field allows the end user to input multiple words (space separated) and will match a row containing those words, even if not in the order that was specified (this allow matching across multiple columns). Note that technically the search is actually a filter, since it is subtractive, removing data from the data set as the input becomes more complex.

- Filtering

Finally the application provides functionality to filter the learning objects by eight available criteria. The eight criteria to filter desired results are: language, language level, education level, education year, difficulty, author, keyword and tag. The filter is hidden by default but can be made visible by clicking on the Filters bar. After specifying the filters and pressing 'Apply filters' only the learning objects that meet the filter criteria will be shown.

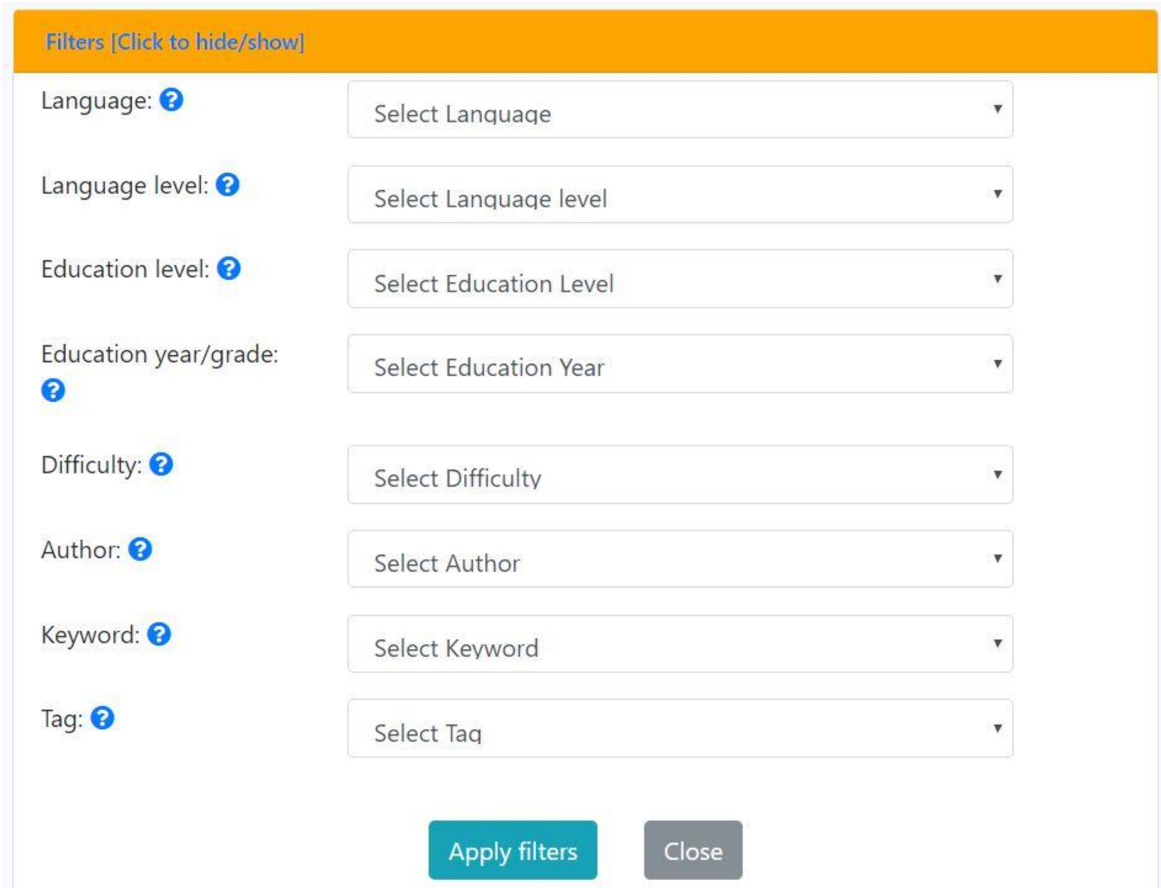


Figure 15 The Learning Objects Filter

The approach presented in this paragraph is our proposed solution for requirement 11: *The system will facilitate searching for a learning object.* The filtering interface presented here will also help in fulfilling the associated usability requirements. The proposed solution in this paragraph will also contribute to meeting usability requirements 1,2 and 3:

- 1: *In each overview (students, learning objects and challenges) a user should be able to sort the results based on different criteria/properties.*
- 2: *In the learning objects overview, the user should be able to filter the wanted results by applying a filter with the desired specification.*
- 3: *In each overview (students, learning objects and challenges) a user should be able to search for the desired object by entering a search term. It should be possible to use incomplete info in the search function.*

7.4.6 Challenge results

A teacher or supervisor should be able to consult the status of a challenge. Because an exhaustive learning analytics will be the subject of a separate module, we limit ourselves here to presenting the status of a challenge. Three statuses are possible for a challenge:

- *Open*: the challenge is offered to the student but he/she has not started it yet
- *Started, but not completed*: the student has started the learning activity, but did not successfully complete all necessary parts
- *Completed*: the challenge has been successfully completed

The teacher or supervisor can check the status of each challenge. To find the desired challenge in the table overview, the user can sort the results by column or use the search field. To get a better insight it is also possible to check the statuses of all challenges offered to a particular student. It is also possible to get an overview of all challenges that contain a certain learning object.

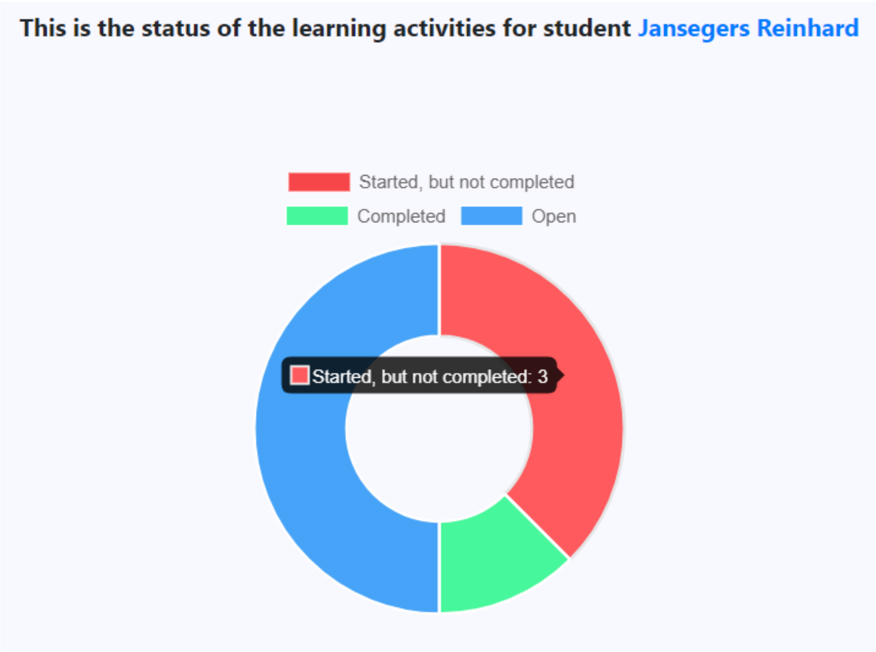


Figure 16 Overview of Challenge Statuses For a Student

The approach presented in this paragraph is our proposed solution for requirement 19: *The user can consult the results of challenges.* This approach also helps to meet the usability requirement nr. 12: *The resulting overview of the challenges for a particular student and the resulting overview of the challenges with a certain learning object should be represented in a visual manner, e.g. by a chart.*

7.5 Summary

In this chapter we started with presenting the technology choices for the authoring tool we have developed in this thesis. We justified the reasons for choosing CodeIgniter 3 as the underlying PHP framework for the back-end. For the front-end we have chosen Twitter Bootstrap 4 in combination with CSS 3 and JQuery. SQLite served as the database technology.

Further, we have discussed the design of our system. In this project we have chosen to use a Model-View-Controller (MVC) architecture. We have then presented our database model that has been structured around the tables for student profiles, learning objects and challenges. These three also formed the core of the design model that was discussed in the section.

Finally, we presented some implementation issues. We discussed how the Xerte learning objects could be made xAPI-ready in order to store the student results in an LRS. In order to do this, we had to add some additional JavaScript code to the Xerte learning object source code. The procedure to make H5P learning objects xAPI-ready also involves adding our own JavaScript code. For the webservices to communicate with the City Card Authoring Environment, we have chosen for an approach using REST instead of SOAP . Because of the flexibility and compactness of JSON, we preferred it over XML as the format to return requested data. We further discussed the Tag Explorer that was developed to assist users to discover the most suitable tags for learning objects and student interests. We also presented the three developed means to facilitate the search for learning objects: filtering by criteria, searching by keyword and sorting the table. The way the challenge results are presented to the user was the final implementation issue we discussed in this chapter.

8

Evaluation

In this chapter the evaluation of the system will be discussed. We have limited ourselves to an evaluation of the usability of the application based on the usability requirements that were defined earlier. We start by describing the purpose and set up of the evaluation, next we will discuss the methodology that has been used. We will then present the results of the evaluation. The chapter will end with a discussion of these results and some recommendations for the UI.

8.1 Evaluation Setup

The purpose of the evaluation was to investigate whether the system is usable for authors who want to create, add, view, edit or delete learning objects and challenges. In this context we define usable as the combination of *the 5 Es*: effective, efficient, engaging, easy to learn and error tolerant (Quesenbery, 2004).

For planning the evaluation we started by choosing the participants. The best scenario is to always have real users as participants in an evaluation. In our case for the category of *normal users* we opted for teachers of a secondary school. The system should indeed be usable for this target group. We asked ten secondary school teachers to participate in the evaluation. Literature showed that a number of five participants is sufficient for detecting the main usability problems (Stone, Jarrett, Woodroffe, & Minocha, 2005). The age of the participants was between 23 and 54 years old.

In practice most of the supervisors will probably also be teacher in the school where they work or they will most likely have worked as a teacher before. Therefore, teachers as participants will probably also be able to reveal shortcomings in the user interface for the supervisor. Since teachers and supervisors will probably use the system on their own, we choose for a user observation based on a single user working alone.

We wanted to collect qualitative data as well as quantitative data. Qualitative data includes comments and opinions, likes and dislikes of the participants. Quantitative data is derived from taking measurements during evaluation. We have chosen to do a cognitive walkthrough while the paricipants where performing task using the think aloud observational technique. After the cognitive walkthrough the users were asked to complete the SUS questionnaire. There also was a short post-session discussion with the participants of the evaluation.

The authoring tool was evaluated by letting the ten teachers use the tool in use case scenarios. The participants were given basic tasks to complete. Background and goal of the experiment was made clear to the participants before they started using the tool. A short introduction to TICKLE and the main workflow diagram of the authoring tool was also given to the participants before the experiment was conducted. This introduction took about 15 minutes in total.

For 7 participants, the evaluation took place at my home. For the other 3 participants the evaluation took place in a classroom at their school. The participants have been informed that a session would take about 45 minutes. A regular laptop, as well as some video and audio equipment were used. The browsers used were Mozilla Firefox and Google Chrome. BB FlashBack Express was used to record the screen and audio of the participant. A click counter was also used.

With BB Flashback Express we could record the screen, i.e. we made a screencast. This would give us the opportunity to look back at the recording and find out where the user found the system difficult to use. It can be a very big help if the note taker couldn't keep up with the user actions during evaluation.

The second additional technology used is the little 'Click counter' program. This program, as the name states, counts the clicks during execution. For some usability requirements the measurement is done by counting the number of interactions with the system.

Each session consisted of:

- a 10-15 minute introduction to TICKLE and explanation about the purpose of the tool
- starting the screen and audio recording with BB FlashBack Express + start Click counter
- 8 tasks executed by participant + cognitive walkthrough questions and note taking by evaluator
- stopping BB FlashBack Express and Click counter
- post-session interview

8.2 Evaluation Methodology

8.2.1 Preparation of the user tasks

The following user tasks were formulated:

Task Doc1: *Adding a given user profile to the system. Editing the user profile to make a small change.*

Try to add a student profile with these data and save:

- First name: Pascal**e**
- Last name: Pieters
- Date of birth: 03/11/1997
- Sex: male

Has the submitting been successful? Do you notice what is wrong?

Now fill in those fields:

- xAPI username: username
- xAPI password: password
- xAPI mbox: myemail
- Submit.

Change the typo in the first name: Pascal instead of Pascale

Task Doc2: *Adding an already created Xerte learning object to the system. Editing the learning object to make a small change to the sensitive content and the keywords.*

On the desktop you will find a Xerte learning object Favelas.zip. Add this learning object to the system with the provided metadata.

- title: favelas
- education level: General Education Level
- education year: 5
- difficulty: difficult
- pedagogic goal: applying

- interactivity level: low

Save. Is this possible? Why (not)?

Add these metadata:

- language: nl
- language level: easy
- sensitive content: drugs
- keyword: urbanization

Save the changes

Task Doc3: *Adding an already created H5P learning object to the system. Editing the learning object to make a small change to the keywords.*

Navigate to the overview of H5P learning objects.

Find the learning object 'Latin words'

Copy the embed code

Add the learning object to the database, using these data:

- title: Latin words
- education level: General Education Level
- education year: 5
- difficulty: difficult
- pedagogic goal: applying
- interactivity level: low
- language: nl
- language level: easy

Save.

Edit the learning object to add 'Latin language' as a keyword.

Task Doc4 : *Using the Tag explorer*

Edit the learning object *Favelas*. Use the tag explorer to add these tags: *sociology*, *urban environment, poverty*.

Task Doc5 : *Creating a challenge*

Add a challenge containing learning object *Favelas* for student Pascal Pieters.

Task Doc6 : *Finding a learning object by sorting, filtering and search*

Navigate to the learning objects overview.

- Find the (alphabetically) last learning object title by sorting.
- Filter out the learning objects in English that are considered difficult.
- Search for all learning objects that contain berrie in their title.

Task Doc7 : *Checking the challenge statuses of a student*

Find how many challenges with status *Open* student Reinhard Jansegers has. How many with status *Started, but not completed*? How many with status *Completed*?

Task Doc8 : *Finding matching learning objects for a student*

Find the three learning objects that are best suitable for student Reinhard Jansegers' interest. Then find the three learning objects that are best suitable for student Reinhard Jansegers' interest and that are *easy*.

8.2.2 Collecting data

Pre-experiment

In the pre-experiment we wanted to collect information about the technical knowledge of the participants. Therefore they were asked how frequently they make use of a computer and if they have any experience with platforms to create interactive online learning objects. They were also asked if they frequently make use of a learning management system at their school.

Cognitive walkthrough questions

In a cognitive walkthrough questions are asked to determine whether a user is likely to recognize what to do with an interface and make appropriate decisions. We used the following cognitive walkthrough questions (Stone et al., 2005):

Question	Technical description	Question to ask in evaluation
1	How does the user know what to do next? Is the correct action sufficiently evident to the user, i.e. recognition instead of recall?	Is there anything there that tells you what to do next?
2	Will the user connect the description of the correct action with what he or she is trying to do?	Is there a choice on screen that lines up with what you want to do? If so, which one?
3	On the basis of the system's response to the chosen action, will the user know if he or she has made a right or wrong choice?	Now that you've tried it, has it done what you wanted it to do?

These questions were asked during the evaluation.

We also asked the participants to think aloud during the task execution. This helped us to follow the reasoning of the evaluator during execution. Comments like '*I am now looking how I can start the editing*' resulted in valuable information for evaluating the usability of the GUI.

During the observation the evaluator took notes. These notes can contain user's remarks made during evaluation or the observer's observations. The answers to the cognitive walkthrough questions were also written down. To structure the note taking, we have split every task up into the necessary user actions to accomplish the task. These eight task descriptions can be found in appendix A.

Conducting post-session discussions

After each evaluation a short post-session interview took place. Users were asked about their thoughts and actions. This gave the participants an opportunity to reflect on their interaction with the system and gave feedback to the evaluator.

Questionnaire

After the evaluation, there was a short questionnaire to complete. We have chosen to use the System Usability Scale (SUS) (Brooke, 1996). The SUS is used to assess the usability, ease of use and learnability of a software system.

When a SUS is used, participants are asked to score the following 10 statements using a Likert scale that ranges from *Strongly Agree* to *Strongly disagree*. The SUS also served as a basis for a short interview afterwards. The SUS we used can be found in appendix B, but we also mention the ten statements here:

- I think that I would like to use this system frequently.
- I found the system unnecessarily complex.
- I thought the system was easy to use.
- I think that I would need the support of a technical person to be able to use this system.
- I found the various functions in this system were well integrated.
- I thought there was too much inconsistency in this system.
- I would imagine that most people would learn to use this system very quickly.
- I found the system very cumbersome to use
- I felt very confident using the system.
- I needed to learn a lot of things before I could get going with this system.

The participants completed the SUS immediately after the cognitive walkthrough of the system to guarantee an outcome that would reflect the user's impression of the system usability.

8.3 Findings

The findings of the evaluation have been based on the cognitive walkthrough notes taken by the observer, the post-session discussions, the SUS questionnaire and the quantitative data collected.

8.3.1 Pre-experiment

In the pre-experiment all participants confirmed that they use the learning management system (LMS) of their school on a (nearly) daily basis. The two oldest participants indicated that they did not consider themselves very computer-minded and that they only knew the basics they frequently use. Two other participants indicated that they had some experience in creating learning objects. One of them had used Bookwidgets, the other one sometimes uses Educaplay and Learningapps. None of the other participants has ever created learning objects, although five out of the remaining six were aware that software to do this exists.

8.3.2 Cognitive Walkthrough and Post-Session Discussions

The cognitive walkthrough and post-session discussions have revealed some usability weaknesses in the tool. We will briefly discuss those here.

1. Importance of the language used in the GUI

The first usability issue we encountered stressed the importance of using the user's native language in the user interface. We had already planned to make translations into Dutch and French, but both translations weren't ready at the start of the evaluation. During the first two evaluation sessions only an English version of the system was available. We noticed a severe impact on the user's performance in executing the task scenarios because both users did not seem to have a good knowledge on the English language. Therefore, we decided to postpone the remaining evaluation sessions until a basic Dutch version was available. The results of both first participants haven't been taken into account in the rest of this usability evaluation.

2. The Tag Explorer

The *Tag Explorer* was developed to assist the user in discovering the best matching tags to specify both student interests and learning object topics. The starting point for the user was to select a language for the tags and a category. These choices would reveal a number of tags that could all be used for revealing more refined, broader or related terms.

In task scenario 4 the eight users were asked to add three specific tags to a learning object: *social sciences, poverty* and *urban development*. From the 24 tags that were added, only six were found inside a minute. For fourteen tags it took over a minute and on four others the user gave up during search. These results were far from satisfactory, since we stated in usability requirement 14 that every user should be able to discover a tag in less than one minute. In the post-session discussion all users indicated that task 4 had been the most difficult. It became clear that a different approach for the *Tag Explorer* should be used.

3. Confirmation of added student profiles, learning objects and challenges

In several task scenarios users were asked to add and edit an object: a student profile (scenario 1), a learning object (scenarios 2 and 3) or a challenge (scenario 5). When the users had pressed *submit* and returned to the main overview, they were asked if they had any indication if the object was added successfully. Only two of the eight users noticed the green confirmation panel above the table. The others started scrolling the table with student profiles, learning objects and challenges in order to find the object they had just added. The results of this task scenario revealed that the confirmation after adding an object should be made clearer to the user.

4. The 'add' button

Users were asked to first add a student profile. Three out of eight users didn't find the *add* button below the table right away. Two of them even left the page to search if the required function would be available on another page. For adding a challenge, two out of eight participants experienced the same problem. In the post-session discussion two participants mentioned that it would be more intuitive to them if the button had been above the table instead of below the table.

5. Validation error messages

Participants were faced with validation errors in task scenarios 1 and 2. For three of the participants the validation errors were immediately clear when they realized that the submitting had failed. Two participants failed to find out which validation errors had occurred in scenario 1, but found them quickly in scenario 2. They had obviously learned from their experience in the first task scenario.

For the three remaining participants however, it took a relatively long time to find that validation errors had occurred. One participant even gave up to find out what had gone wrong. Even when it was clear there were validation errors, these last three participants had difficulty finding all the necessary fields they had failed to complete. They pressed the submit button several times, thinking they had completed every required field, just to find out something was still missing. From these results, it became clear that the validation errors should be made clearer to the user.

6. Sorting the table by column

In task scenario 6 the participants were asked to sort the table by title in descending order to find the (alphabetically) last learning object. Six out of eight participants did not manage to find that clicking the column header 'Title' would accomplish this task. They either had not noticed the arrows up and down for sorting or were not familiar with these symbols in a table context. We must however point out here that the possibility of sorting was not mentioned during the introduction...

7. Other remarks, hints and tips

From the post-session discussions with the participants and during the thinking aloud performed during the execution of the tasks scenarios, a number of valuable remarks, hints and tips were given:

- The filter text should be clearer or have a larger font.
- The icons for viewing (magnifying glass), editing (pencil) and delete (trash bin) should have a tooltip text because the user was not familiar with those icons.

- In an overview table, the objects should be ordered by the time they were added (in descending order) so that the most recent objects appear on top.
- In the pie chart with the status results (*open*, *started*, *completed*) the absolute number for each status can be found by hovering the pie chart. One participant mentioned this could be made clearer by placing these numbers for all three possible statuses also next to the pie chart.
- When selecting a Xerte.zip-file for upload, one participant mentioned this could be facilitated by only showing .zip files when selecting from the local system. He also asked if it would be possible to add the size of the file in the confirmation that the file is ready for upload.
- When adding or editing a student profile or a learning object the participants had to enter the language(s). One participant noticed that it might be more appropriate to put the languages that are likely to be used most on top of the drop down list, instead of in the alphabetical order that was provided.
- One participant indicated that, besides the drag and drop possibility to select tags, a double-click to select and deselect them would be an enhancement.
- As for the eight available filter criteria in the tool, an option to clear all filters was proposed by two participants.

8.3.3 Evaluation Findings of Selected Usability Requirements

In this paragraph we discuss the evaluation of some usability requirements that were formulated in section 6.2.3.

The first three usability requirements aim to facilitate finding the desired row in a table overview with students, learning objects or challenges. In order to achieve this a user can sort, search or filter the data. All participants had noticed the search field very quickly. Apart from one participant, there also was not any hesitation when users were asked to filter the data with the given criterion. Only two participants however managed to find the sorting function in the table headers.

All participants managed to find the on-screen location where they could visit the details of students or learning objects (requirement 4). Two more hesitant participants asked for a tooltip text to make sure they would be visiting the correct link. The four-step process to create challenges was immediately clear to all participants (requirement 10). As stated above, all participants struggled to add the tags given in the task scenario within the time frame of one minute (requirement 14). Only six of the 24 tags were found within 60 seconds. This was a clear indication that the Tag Explorer's usability needed improvement.

8.3.4 The SUS Questionnaire

The SUS Questionnaire was filled in by eight participants.The results are represented in Figure 17 and Figure 18.

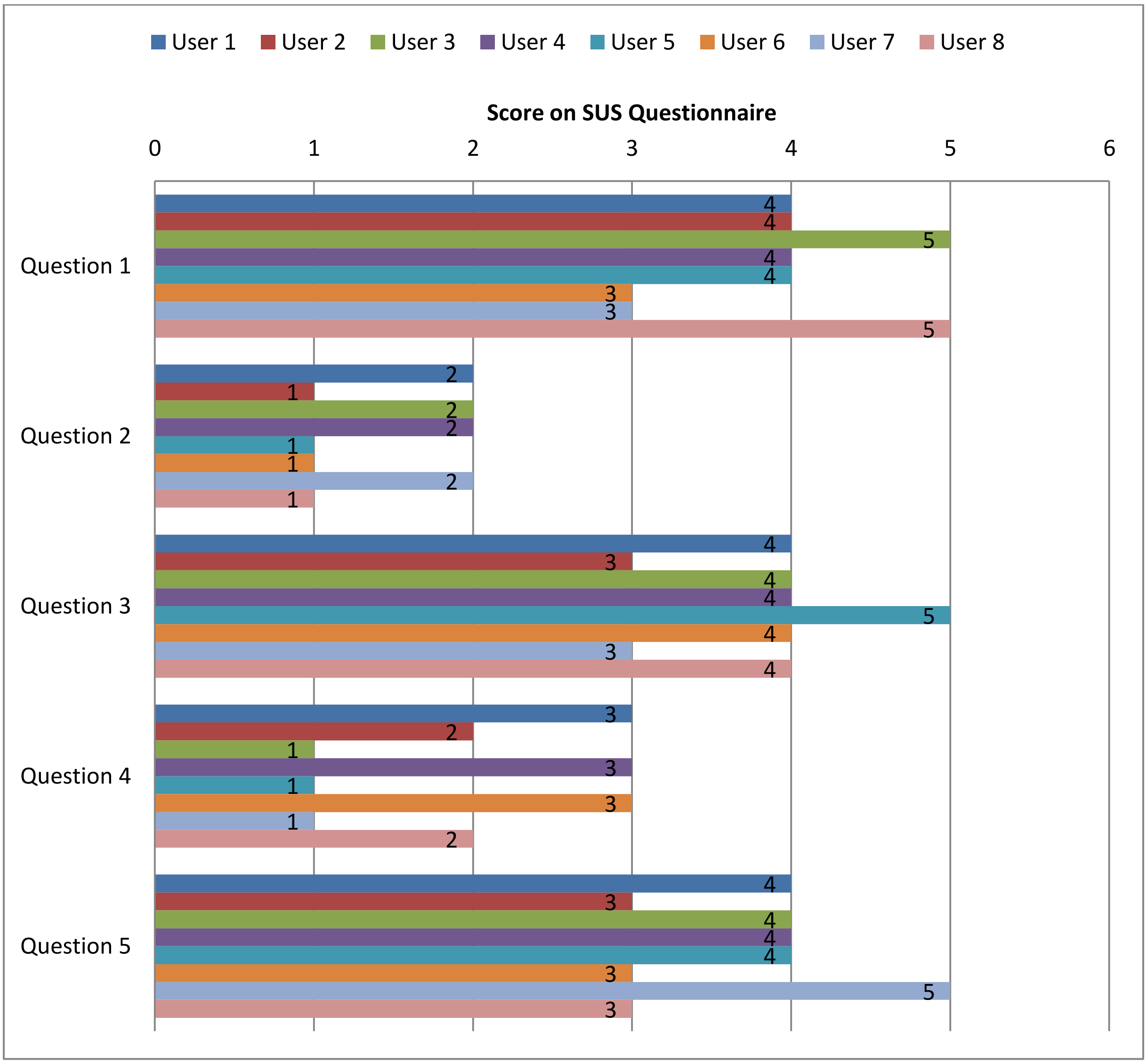


Figure 17 Results of the SUS Questionnaire for questions 1 to 5

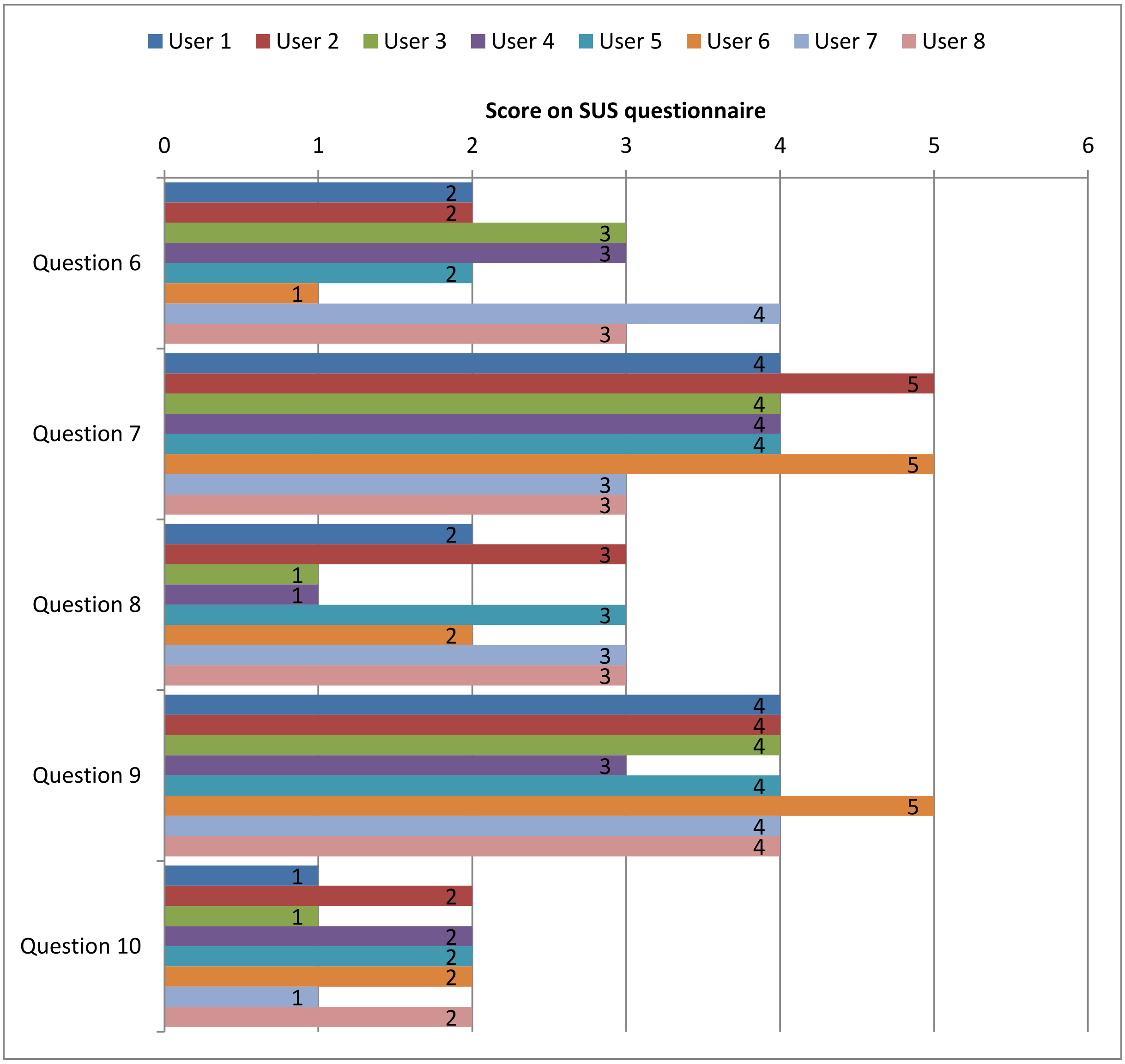


Figure 18 Results of the SUS Questionnaire for questions 6 to 10

To calculate the usability scores we scale down all results to the 0-4 score interval. In order to do so, we subtract one from the user response for the odd numbered questions. All these question are positively formulated. Therefore a higher score implicates better usability.By subtracting one from the user response, we map the scores from the users from the 1-5 interval to the 0-4 interval. For the even numbered questions we have to take a different approach, since these questions have a negative formulation, where higher scores are an indication of lower system usability. We therefore have to reverse the scores given by the participants, so we subtract the user response from 5. With this operation we get scores for the even numbered questions in the 0-4 interval, where higher scores imply better system usability. The total score percentage per user can then be calculated by adding all 10 scores and multiplying by 2.5 to obtain a score on 100. Table 8 shows the total scores that were calculated. The scores are between 67,5 and 82,5.

User	1	2	3	4	5	6	7	8
Total	**75**	**72,5**	**82,5**	**70**	**80**	**77,5**	**67,5**	**70**

Table 8 Total Scores of the Participants

It has been demonstrated that the usability of applications with a total score below 55 is considered not acceptable, while scores above 70 indicate that the usability is acceptable (Bangor, Kortum, & Miller, 2009). This implicates that for only one out of eight participants the usability of the tool in the grey zone between not acceptable and acceptable. This participant was the oldest in the panel. She also told us in the post-session discussion that she is not very computer-minded. For all other participants the usability of the system is acceptable.

8.4 Results

The evaluation results have led to a number of adaptations to improve the usability of the tool.

A first major improvement has been done to the *Tag Explorer*. Here we have added a possibility to search for tags in a search field. The *Tag Explorer* will then show all tags that are available in the system and contain the text in the search field. This will provide the user much quicker with a relevant starting point for tag discovery. Besides the search field we have also implemented the suggestion given by one of the participants to select and deselect tags by double-clicking.

A second improvement has been the addition of a confirmation that a student, learning object or challenge has been added. We have prolonged the time that the (green) confirmation message remains on top of the page from five seconds to fifteen seconds and have added a short wobble animation to this item. It has been demonstrated that the use of animation attracts user attention and facilitates quicker location of the animated target item in screen displays.(Hong, Thong, & Tam, 2004). We also make sure that the overview table on the next page is ordered by date in descending order. This way the newly added element occupies the first row. To make this even more clear, we have added a small green label with the text 'New' in the first column of the newly added element. These three measures should ensure the user will be aware of a successful add operation.

As suggested by the participants who could not find the *Add* button, we have transferred this button from the bottom of the page to the top, where the users expected it to be. As for the validation errors that were not immediately clear to a number of users, we have now added an animated error message at the top of the page where the user can see how many fields aren't properly completed. The user will then be aware of how many red input fields with error messages he should look for.

Finally, we also implemented some of the valuable tips that were given by different participants: we added a tooltip to the icons for viewing, editing and deleting an object, we made sure that only .zip files are visible to choose from the local system when selecting a Xerte object for upload, in the drop down lists with languages we have put the most common options (nl, fr, en-GB) on top, we have added a button to clear all filters...

One last improvement was something we noticed ourselves during the execution of the task scenarios. Filters can also be used to filter table rows by keyword or tag. This is possible by selecting the keyword of tag from a drop down list in alphabetical order, containing all keywords and tags that have been used at least once. Since our demo information for evaluation purposes was limited, the size of the drop down lists with tags and keywords was not very large either. Users could find the desired filter for tags or keywords easily. However, if the list of used keywords and tags increases drastically, finding the right tag or keyword would be much more difficult. Therefore, we have added a live search to the select tag, so the user can filter out the options containing what is entered in the live search field.

Note that after these usability improvements the tool has not been evaluated again. We have shown the improvements to two participants. In this informal contact, both participants were very positive about the changes.

8.5 Summary

In this chapter we have presented the usability evaluation of the authoring tool that was developed for this thesis. Ten secondary high school teachers were the participants for the evaluation. For the first two participants only an English version of the authoring tool was available. Because both users didn't have a very good knowledge of the English language, we noticed a severe impact on the user's performance. Therefore, we made a Dutch translation of the tool. This version was available for the next eight participants. Only their results have been taken into account.

For the evaluation we formulated eight user tasks. During the execution of the tasks the participants were encouraged to think aloud and they were posed cognitive walkthrough questions. After the user tasks, the participants completed the SUS Questionnaire and a short post-session interview took place.

The evaluation revealed a number of usability weaknesses. Users weren't able to discover the correct tags using the *Tag Explorer*. Since the problem seemed to be the starting point, we also added the possibility to search the available tags in a search field to tackle this deficit. Further, the confirmation and error messages were not found by all users. Therefore we have added animation and focus to these messages whenever they appear on screen. We also added a green label to newly added objects. Besides these changes, we also received a number of valuable hints, tips and remarks concerning usability. All of them have been taken into account in the latest version of the authoring tool.

From the results of the SUS Questionnaire we concluded that all participants, except one, found the system usability acceptable to good.

9
Conclusions

9.1 Summary

The purpose of this thesis was to create an authoring tool for casual users to steer the personalized delivery of learning activities in a mobile web 2.0 environment. We have started our work by introducing the TICKLE project that serves as the context of the tool. After giving a motivation for the development of the tool, we have set out the research goals for this project.

In chapter 2, we have discussed common technologies and standards for tracking results of learning objects and compared SCORM and xAPI/Tin Can. This comparison was performed to justify the choice for xAPI over SCORM in our authoring tool.

In chapter 3 we have researched related work, i.e. existing authoring tools for the creation of learning activities. We have summarized the main features of the most widespread free and open source solutions. We have also briefly discussed a number of well-known commercial products. The study of the related work in this chapter resulted in a choice for supporting two existing authoring tools in our authoring tool: H5P and Xerte.

In chapter 4 we have done a literature study in order to determine which information to include in the student profile. Information found in the three background research studies that were done by Vlieghe in the context of the TICKLE project was combined with information about models used in the field of adaptive e-learning, mainly regarding the learning style and preferences of the student.

In chapter 5 we investigated the representation of the learning object metadata. We identified different models for metadata representation, IEEE LOM being the most commonly used in the field of education. We also explained the use of application profiles that only use a subset of all elements available in IEEE LOM. After discussing different application profiles, we explained our choice for the Learning Resource Exchange Metadata Application Profile (LREMAP) and defined the subset of elements from IEEE LOM suitable for our purposes.

In chapter 6 we formulated the requirements for the authoring tool, both functional and non-functional. Since the system must be usable for casual users, i.e. users without computer science background, special emphasis was put on the usability requirements. Use case scenarios for these usability requirements were designed.

Chapter 7 started with the technological choices we made for the system, followed by the design of our solution. The database model, as well as a design class model was given. Next, we elaborated on different concerns in the implementation: making learning objects xAPI ready and providing access to the data through REST APIs. Further, we also discussed the implementation of the Tag Explorer to assist the user in discovering the right tags and the Matcher for the discovery of learning objects that fit a student's interest.

The developed system was evaluated by ten teachers from a secondary school. The setup, methodology and results of the evaluation were presented in chapter 8. The evaluation results yielded several adaptations to the tool to enhance usability. In general however, seven out of eight participants considered the usability of the tool (at least) acceptable.

9.2 Future Work

In this section we will discuss possible improvements that could be made to the tool. We will also propose additional functionality to extend the system.

- **Extending the collected tracking information**

At the moment the only information that is stored in the LRS is the status of a challenge. A challenge has one of three possible statuses: *open*, *started* or *completed*. The status is changed from *started* to *completed* when the youngster has passed all activities in the learning object. We could extend the system to send additional data to the LRS to support the learning analytics module that will be developed at a later stage. The data to collect could be as detailed as needed: score for each activity in the learning object, time spent on each page, number of clicks, number of correct answers, number of wrong answers,... . This will require adding additional code to the learning objects when they are added to the tool.

- **Recording other learning experiences**

With the arrival of xAPI, learning is no longer bound to a LMS. xAPI opens up a lot of additional possibilities. We could extend the tool with a module to let the youngster record his/her learning experiences that are outside the scope of traditional learning. A visit to a museum, a concert or a play, an interesting article or book that has been read: it could all be recorded by the youngster and saved in the personal LRS. The youngster could then be rewarded for this non-conventional learning.

- **Integrating Xerte and H5P**

Xerte Online Toolkits and H5P are self hosted solutions to create learning objects. They consist of PHP code that can be hosted on our own server. The steps for creating and importing the learning objects into the authoring tool are now as follows:

  - the author creates the learning object in Xerte or H5P
  - the author exports and downloads the Xerte learning object to his own computer; for H5P he copies the embed code
  - the author adds the learning object to the authoring tool

This procedure for an author could be simplified and reduced to these steps:

  - the author creates the learning object in Xerte or H5P within the authoring tool
  - when the author saves the learning object he can immediately complete the metadata to add it to the system

To achieve this, we will need to integrate the Xerte Online Toolkits and H5P in the authoring tool. Both have no API's we could use for this, but since Xerte and H5P are open source software, we could alter the source code to accomplish the reduced workflow. If we add an extra button to the Xerte and H5P GUI, the user could save the learning object without the need for extra steps. For Xerte the export and upload of the learning object will then happen in the background. For H5P the generated embed code will be copied without the user having to know.

- **Supporting other learning object authoring tools**

For now learning objects can be created with Xerte Online Toolkits and H5P. Both are free open source solutions. In the future it might be possible to also consider commercial authoring tools like Bookwidgets and integrate them into the project. If it is clear that authors have a certain preference for an existing authoring tool to create learning objects, this should be supported.

- **Translation**

At the time of writing (May 2018) a complete English version of the tool is available, as well as a partial translation in Dutch. As mentioned in chapter 8, the Dutch translation was made to overcome the language barrier for the participants in the usability evaluation. For the translated version, the focus was on the screens that would appear in the task scenarios of the evaluation. The options in the select tags that are read from the database (difficulty levels, nationalities, sensitive content, topic categories ...) aren't translated into Dutch yet. Because the tool needs to be used in the Brussels-Capital region, it will not be sufficient to finish the Dutch translation; a French translation will also be required. In addition, the language should be stored in the user profile.

- **Recommendations**

As the learning objects library will grow, it will become harder for teachers to select the right learning objects to create challenges. For this purpose a true recommendation system could be integrated into the system. Another possible extension could be the possibility to leave a *review* of a learning object that was used in a challenge. One's opinion could be recorded in a star rating (1 to 5) and a short comment. This way good quality learning objects might get more attention.

References

Abdullah, M., Daffa, W. H., Bashmail, R. M., Alzahrani, M., & Sadik, M. (2015). The Impact of Learning Styles on Learner's Performance in E-Learning Environment. *IJACSA) International Journal of Advanced Computer Science and Applications*, *6*(9)

Adapt. (2016). Retrieved 12 April 2018, from https://www.adaptlearning.org/

ADL. (2017). Tin Can API Registry. Retrieved 15 November 2017, from https://registry.tincanapi.com/#home/activityTypes

ADLNET. (2017a). SCORM Overview. Retrieved 8 May 2017, from http://adlnet.gov/adl-research/scorm/

ADLNET. (2017b). xAPI Architecture Overview. Retrieved 3 May 2017, from https://www.adlnet.gov/adl-research/performance-tracking-analysis/experience-api/xapi-architecture-overview/

ADLNet. (2017). xAPI Evolution. Retrieved 7 May 2017, from http://experienceapi.com/tin-can-evolution/

ADLNET. (2018). ADL_LRS (Version 1.0.3.). Retrieved from https://github.com/adlnet/ADL_LRS

Adobe Captivate. (2018). Retrieved 13 April 2018, from http://www.adobe.com/be_en/products/captivateprime.html?promoid=7JJ16KCZ&mv=other

Apereo. (2016, August 15). OpenLRS. Retrieved 7 August 2017, from https://github.com/Apereo-Learning-Analytics-Initiative/OpenLRS/issues

Apereo. (2018). Retrieved 13 April 2018, from https://www.apereo.org/

Articulate Storyline 360. (2018). Retrieved 13 April 2018, from https://articulate.com/360/storyline

Authoring system. (2017). In *Wikipedia*. Retrieved from https://en.wikipedia.org/w/index.php?title=Authoring_system&oldid=800701764

Bangor, A., Kortum, P., & Miller, J. (2009). Determining what individual SUS scores mean: Adding an adjective rating scale. *Journal of Usability Studies*, *4*(3), 114–123.

Barber, W. (1998). Culturability: The merging of culture and usability. In: J. Cantor (Ed.) *4th Conference on Human Factors and the Web* (pp. 2–10). Basking Ridge,USA.

Barker, P. (2005). What is ieee learning object metadata/ims learning resource metadata. *CETIS Standards Briefing Series, JISC (Joint Information Systems Committee of the Universities' Funding Councils).*

Berking, P. (2015). *Choosing a learning record store (LRS).* November, retrieved from http://adlnet.gov/adlassets/uploads/2015/11/Choosing-an-LRS.pdf

BrightCookie. (n.d.). Retrieved 11 April 2018, from http://www.brightcookie.com/what-we-do/

Brooke, J. (1996). SUS-A quick and dirty usability scale. *Usability Evaluation in Industry*, *189*(194), 4–7.

Brown, E., Brailsford, T., Fisher, T., Moore, A., & Ashman, H. (2006). Reappraising cognitive styles in adaptive web applications. In: L. Carr, D. De Roure, C. Goble and M. Dahlin (Eds.) *Proceedings of the 15th international conference on World Wide Web* (pp. 327–335). ACM.

Brusilovsky, P., & Millán, E. (2007). User models for adaptive hypermedia and adaptive educational systems. In P. Kobsa & W. Nejdl, *The adaptive web* (pp. 3–53). Springer-Verlag.

Callahan, E. (2005). Cultural similarities and differences in the design of university web sites. *Journal of Computer-Mediated Communication*, *11*(1), 239–273.
Calle, D. O. D. L. (2007). *Realizing interoperability of e-learning repositories* (http://purl.org/dc/dcmitype/Text). Universidad Autónoma de Madrid. Retrieved from https://dialnet.unirioja.es/servlet/tesis?codigo=19692
Casali, A., Deco, C., Romano, A., & Tomé, G. (2013). An assistant for loading learning object metadata: An ontology based approach. In: D. Guralnick, J. Kinnamon and S. Schreiter (Eds.) *Interdisciplinary Journal of E-Learning and Learning Objects* (pp. 77-87). Informing Science Institute
Chan, T. T., & Bergen, B. (2005). Writing direction influences spatial cognition. In: B. Bara, L. Barsalou and M. Bucciarelli (Eds.) *Proceedings of the 27th annual conference of the cognitive science society* (pp. 412–417). Stresa, Italy: Lawrence Erlbaum.
Chattopadhyay, A., Darke, P. R., & Gorn, G. J. (2002). Roses are red and violets are blue-everywhere? Cultural differences and universals in color preference and choice among consumers and marketing managers. Retrieved from https://papers.ssrn.com/sol3/papers.cfm?abstract_id=340501
Chen, Y.-F., Zhao, H.-K., Yu, X.-Q., & Wan, W.-G. (2010). Research on Ontology-based User Interest Model Construction [J]. *Computer Engineering*, *21*, 017.
Chirita, P. A., Nejdl, W., Paiu, R., & Kohlschütter, C. (2005). Using ODP metadata to personalize search. In: R. Baeza-Yates and N. Ziviani (Eds.) *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 178–185). Salvador, Brazil: ACM.
Conlan, O., Dagger, D., & Wade, V. (2002). Towards a standards-based approach to e-Learning personalization using reusable learning objects. In: M. Driscoll and T.C. Reeves (Eds.) *Proc. of World Conference on E-Learning, E-Learn* (pp. 15–19).
Cristea, A., & Calvi, L. (2003). The three layers of adaptation granularity. In: P. Brusilovski, P. Corbett and F. de Rosis (Eds.) *International Conference on User Modeling* (pp. 4–14). Johnstown, USA: Springer.
Cyr, D., & Trevor-Smith, H. (2004). Localization of Web design: An empirical comparison of German, Japanese, and United States Web site characteristics. *Journal of the Association for Information Science and Technology*, *55*(13), 1199–1208.
Daniel, A. O., Oludele, A., Baguma, R., & Weide, T. (2011). Cultural issues and their relevance in designing usable websites. Retrieved from http://repository.ubn.ru.nl/bitstream/handle/2066/91681/91681.pdf
Dolog, P., Henze, N., Nejdl, W., & Sintek, M. (2004). Personalization in distributed e-learning environments. In: S. Feldman and M. Uretsky (Eds.) *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters* (pp. 170–179). ACM.
Doulik, P., Skoda, J., & Simonova, I. (2017). Learning Styles in the e-Learning Environment: The Approaches and Research on Longitudinal Changes. *International Journal of Distance Education Technologies (IJDET)*, *15*(2), 45–61.
Dunn, R. (2003). The Dunn and Dunn learning style model and research. *New York: St Johns University*.
Duval, E., & Hodgins, W. (2003). A LOM Research Agenda. In *WWW (Alternate Paper Tracks)*.
Elucidat. (2017). Retrieved 14 April 2018, from https://www.elucidat.com/
EUR-Lex - 52010DC2020 - EN (2010). Retrieved from http://eur-lex.europa.eu/legal-content/en/ALL/?uri=CELEX%3A52010DC2020
Europea, C. (2013). Reducing early school leaving: Key messages and policy support. *Final Report of the Thematic Working Group on Early School Leaving). Bruxelles*.

Eurostat. (2016). Early leavers from education and training - Statistics Explained. Retrieved from http://ec.europa.eu/eurostat/statistics-explained/index.php/Early_leavers_from_education_and_training

eXeLearning. (2018). Retrieved 14 April 2018, from http://exelearning.net/?lang=en

Felder, R. M., Silverman, L. K., & others. (1988). Learning and teaching styles in engineering education. *Engineering Education*, *78*(7), 674–681.

Felder, R. M., & Soloman, B. A. (1999). Index of learning styles (ILS). *On-Line at Http://Www2. Ncsu. Edu/Unity/Lockers/Users/f/Felder/Public/ILSpage. Html*.

Fleming, N. D. (1995). I'm different; not dumb. Modes of presentation (VARK) in the tertiary classroom. In *Proceedings of the 1995 Annual Conference of the Higher Education and Research Development Society of Australasia (HERDSA), HERDSA* (Vol. 18, pp. 308–313). Rockhampton, Queensland.

Friesen, N. (2004). Final report on the" International LOM Survey. *ISO/IEC JTC1 SC36 N*, *871*, 2004.

Friesen, N., Roberts, A., & Fisher, S. (2002). Cancore: Metadata for learning objects. *Canadian Journal of Learning and Technology/La Revue Canadienne de l'apprentissage et de La Technologie*, *28*(3).

Gassler, G., Hug, T., & Glahn, C. (2004). Integrated Micro Learning–An outline of the basic method and first results. *Interactive Computer Aided Learning*, *4*, 1–7.

Gluz, J. C., Silveira, E. L., Da Silva, L. R. J., & Barbosa, J. L. V. (2016). Towards a Semantic Repository for Learning Objects: Design and Evaluation of Core Services. *J. UCS*, *22*(1), 16–36.

Gomo learning. (n.d.). Retrieved 12 April 2018, from https://www.gomolearning.com/

GrassBlade. (n.d.). Retrieved 13 November 2017, from http://www.nextsoftwaresolutions.com/grassblade-lrs-experience-api/

H5P. (n.d.). Retrieved from https://h5p.org/

Hall, E. T., Hall, M. R., & others. (1989). *Understanding cultural differences*. Intercultural press. Retrieved from http://agris.fao.org/agris-search/search.do?recordID=US201300663324

Han, X., Shen, Z., Miao, C., & Luo, X. (2010). Folksonomy-Based Ontological User Interest Profile Modeling and Its Application in Personalized Search. In: An A., Lingras P., Petty S., Huang R. (Eds.) *Proceedings of the 6th international conference on Active media technology* (pp. 34–46). Toronto, Canada: Springer.

Hofstede, G. (2003). *Culture's consequences: Comparing values, behaviors, institutions and organizations across nations*. Newbury Park, CA: Sage publications.

Honey, P., & Mumford, A. (1986). *Using your learning styles*. Peter Honey Maidenhead.

Honey, P., & Mumford, A. (2006). *Learning styles questionnaire: 80-item version*. Maidenhead.

Hong, W., Thong, J. Y., & Tam, K. Y. (2004). Does animation attract online users' attention? The effects of flash on information search performance and perceptions. *Information Systems Research*, *15*(1), 60–86.

HT2. (2018). Jisc - Open Learning Analytics Framework. Retrieved 18 October 2017, from https://www.ht2labs.com/about/clients/jisc-open-learning-analytics-framework/

IMS Global Learning Consortium. (2006). IMS Meta-data Best Practice Guide for IEEE 1484.12. 1-2002 Standard for Learning Object Metadata. *Available on Line Http://Www. Imsglobal. Org/Metadata/Mdv1p3/Imsmd_bestv1p3. Html*.

Ismail, D. M. M., Yin, M., Theng, Y. L., Goh, D. H.-L., & Lim, E.-P. (2003). Towards a role-based metadata scheme for educational digital libraries: A case study in Singapore. *Lecture Notes in Computer Science*, 41–51.

ISO/IEC 9126. (2017). In *Wikipedia*. Retrieved from https://en.wikipedia.org/w/index.php?title=ISO/IEC_9126&oldid=813539588

iSpring. (2017). Retrieved 15 April 2018, from http://www.ispringsolutions.com/#%2
iSpring Solutions. (2014). SCORM or xAPI: Which to Choose and Why. Retrieved 9 May 2017, from http://www.ispringsolutions.com/blog/scorm-or-tin-can/
Jegatha Deborah, L., Baskaran, R., & Kannan, A. (2014). Learning styles assessment and theoretical origin in an E-learning scenario: a survey. *Artificial Intelligence Review*, 1–19.
Kim, H. R., & Chan, P. K. (2003). Learning implicit user interest hierarchy for context in personalization. In: D. Leake (Ed.) *Proceedings of the 8th international conference on Intelligent user interfaces* (pp. 101–108). ACM.
Kim, J., Lee, A., & Ryu, H. (2013). Personality and its effects on learning performance: Design guidelines for an adaptive e-learning system based on a user model. *International Journal of Industrial Ergonomics*, *43*(5), 450–461.
Kolb, D. (1981). Learning styles and disciplinary differences. *The Modern American College*, *1*, 232–255.
Kolb, D. A. (1985). Kolb's learning style inventory. *McBer & Company, Boston, MA*.
Kondratova, I., & Goldfarb, I. (2007). Color your website: Use of colors on the web. In: N. Aykin (Ed.) *International conference on usability and internationalization* (Vol. 1, pp. 123–132). Beijing, China: Springer.
Kumar, V., Nesbit, J., Winne, P., Hadwin, A., Jamieson-Noel, D., & Han, K. (2007). Quality rating and recommendation of learning objects. *E-Learning Networked Environments and Architectures*, 337–373.
Learning Locker. (2018). Retrieved 14 November 2017, from https://learninglocker.net/
Learning Record Store: What is an LRS? (2011). Retrieved 2 June 2017, from https://scorm.com/what-is-an-lrs-learning-record-store/
Learning Record Stores. (n.d.). Retrieved 3 November 2017, from http://www.beyondlms.org/pages/lrs/
Learning Styles. (2014). Retrieved 6 January 2018, from http://www.learningstyles.net/
Learning Tools. (2014). Retrieved 11 December 2017, from http://www.learningtools.arts.ubc.ca/index.html
Lehman, R. (2007). Learning object repositories. *New Directions for Adult and Continuing Education*, *2007*(113), 57–66.
LMS: Learning Management System Online Training Software. (2018). Retrieved 15 October 2017, from https://www.learnupon.com/
Ma, Z., Pant, G., & Sheng, O. R. L. (2007). Interest-based personalized search. *ACM Transactions on Information Systems (TOIS)*, *25*(1), 5.
Magazine, D.-L. (2010). Taming the metadata beast: ILOX. *D-Lib Magazine*, *16*(11/12).
Malta, M. C., & Baptista, A. A. (2014). A panoramic view on metadata application profiles of the last decade. *International Journal of Metadata, Semantics and Ontologies*, *9*(1), 58–73.
Massart, D., Shulman, E., & Van Assche, F. (2011). Learning Resource Exchange Metadata Application Profile version 4.7. *European Schoolnet*. Retrieved 3 December 2017, from http://lreforschools.eun.org/c/document_library/get_file?p_l_id=10970&folderId=12073&name=DLFE-1.pdf
Massart, David. (2009). Towards a pan-European learning resource exchange infrastructure. *Next Generation Information Technologies and Systems*, 121–132.
McClelland, M. (2003). Metadata standards for educational resources. *Computer*, *36*(11), 107–109.
MERLOT II. (2018). Retrieved 11 April 2018, from https://www.merlot.org/merlot/index.htm
MIT OpenCourseWare. (2018). Retrieved 12 April 2018, from https://ocw.mit.edu/courses/find-by-topic/

Mitchell, T., Chen, S. Y., & Macredie, R. (2004). Adapting hypermedia to cognitive styles: is it necessary. In: L.M. Aoryo and C. Tasso (Eds.) *Proc. of Workshop on Individual Differences in Adaptive Hypermedia at the 3rd International Conference on Adaptive Hypermedia and Adaptive Web-based Systems*.

Myers, I. B., McCaulley, M. H., & Most, R. (1985). *Manual: A guide to the development and use of the Myers-Briggs Type Indicator* (Vol. 1985). Consulting Psychologists Press Palo Alto, CA.

OAsis. (2018). Retrieved 17 April 2018, from http://oasis.col.org/

Ochoa, X., & Duval, E. (2006). Use of contextualized attention metadata for ranking and recommending learning objects. In: E. Duval, J. Najjar and M. Wolpers (Eds.) *Proceedings of the 1st international workshop on Contextualized attention metadata: collecting, managing and exploiting of rich usage information* (pp. 9–16). ACM.

Open Directory Project. (2018). Retrieved 10 November 2017, from http://www.opendirectoryproject.org/

OpenStax CNX. (2018). Retrieved 12 April 2018, from https://cnx.org/browse

Papanikolaou, K. A., Grigoriadou, M., Kornilakis, H., & Magoulas, G. D. (2001). INSPIRE: An intelligent system for personalized instruction in a remote environment. In: P. De Bra, P. Brusilovsky and A. Kobsa (Eds.) *Workshop on Adaptive Hypermedia* (pp. 215–225). Sonthofen, Germany: Springer.

Paredes, P., & Rodríguez, P. (2002). Considering sensing-intuitive dimension to exposition-exemplification in adaptive sequencing. In: P. De Bra, P. Brusilovky and R. Conejo (Eds.) *International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems* (pp. 556–559). Springer.

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, *24*(3), 45–77.

Quesenbery, W. (2004). Balancing the 5Es of usability. *Cutter IT Journal*, *17*(2), 4–11.

Reinecke, K., & Bernstein, A. (2013). Knowing What a User Likes: A Design Science Approach to Interfaces that Automatically Adapt to Culture. *Mis Quarterly*, *37*(2).

RELOAD Project. (2008). Retrieved 3 March 2018, from http://www.reload.ac.uk/

Riptide Elements. (n.d.). Retrieved 7 April 2018, from https://learning.riptidesoftware.com/

Rustici Software. (2018a). Experience API Client Libraries. Retrieved 12 October 2017, from http://tincanapi.com/libraries/

Rustici Software. (2018b). Experience API History. Retrieved 12 October 2017, from http://experienceapi.com/history/

Rustici Software. (2018c). Experience API Overview. Retrieved 13 October 2017, from http://experienceapi.com/overview/

Rustici Software. (2018d). Hosted LRS. Retrieved 13 November 2017, from http://tincanapi.com/hosted-lrs/

Rustici Software. (2018e). Layer 1: A Modernized Version of SCORM - Experience API. Retrieved 5 October 2017, from http://experienceapi.com/layer-1-freeing-us-from-the-constructs-of-old/

Rustici Software. (2018f). Layer 3: Free the Data - Experience API. Retrieved 5 October 2017, from http://experienceapi.com/layer-3-free-the-data/

Rustici Software. (2018g). Layer 4: Correlate Job Performance with Training - Experience API. Retrieved 7 October 2017, from http://experienceapi.com/layer-4-correlate-job-performance-with-training/

Rustici Software. (2018h). SCORM Manifest XML Schema Definition Files. Retrieved 6 May 2017, from http://scorm.com/scorm-explained/technical-scorm/content-packaging/xml-schema-definition-files/

Rustici Software. (2018i). SCORM vs xAPI. Retrieved 8 May 2017, from http://experienceapi.com/scorm-vs-the-tin-can-api/
Rustici Software. (2018j). Statement Explorer. Retrieved 12 November 2017, from http://tincanapi.com/statement-explorer/
Rustici Software. (2018k). Statements 101. Retrieved 12 November 2017, from http://tincanapi.com/statements-101/
Sacco, M., Smits, W., Kavadias, D., Spruyt, B., & Andrimont, C. (2016). BSI synthesenota. De Brusselse jeugd: tussen diversiteit en kwetsbaarheid. *Brussels Studies*, *98*, 1–21.
Sharable Content Object Reference Model. (2017). In *Wikipedia*. Retrieved from https://en.wikipedia.org/w/index.php?title=Sharable_Content_Object_Reference_Model&oldid=774060900
Siala, H., O'Keefe, R. M., & Hone, K. S. (2004). The impact of religious affiliation on trust in the context of electronic commerce. *Interacting with Computers*, *16*(1), 7–27.
Sieg, A., Mobasher, B., & Burke, R. (2007). Web search personalization with ontological user profiles. In: A.O. Falcão and Ø.H. Olsen (Eds.) *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management* (pp. 525–534). ACM.
Skillaware. (n.d.). SkillAnalyzer. Retrieved 2 April 2018, from http://skillaware.com/en/skillanalyzer/
SmartBuilder. (n.d.). Retrieved 3 April 2018, from http://www.smartbuilder.com/
Steunpunt tot Bestrijding van Armoede, Bestaansonzekerheid en Sociale Uitsluiting. (2016). Feiten & cijfers. Retrieved 14 June 2017, from http://www.armoedebestrijding.be/cijfers_leefloon.htm
Stone, D., Jarrett, C., Woodroffe, M., & Minocha, S. (2005). *User interface design and evaluation*. Morgan Kaufmann.
tf–idf. (n.d.). In *Wikipedia*. Retrieved from https://en.wikipedia.org/w/index.php?title=Tf%E2%80%93idf&oldid=808154546
TICKLE research project. (2014). Retrieved 2 April 2017, from https://wise.vub.ac.be/tickle/
Triantafillou, E., Pomportsis, A., & Demetriadis, S. (2003). The design and the formative evaluation of an adaptive educational system based on cognitive styles. *Computers & Education*, *41*(1), 87–103.
Trompenaars, F., & Hampden-Turner, C. (2011). *Riding the waves of culture: Understanding diversity in global business*. Nicholas Brealey Publishing.
Vlieghe, J. (2014). Report D2: State-of-the-art on Early School Leaving and Dropouts - TICKLE-Report-D2_final.pdf. Retrieved from https://wise.vub.ac.be/tickle/wp-content/uploads/2015/12/TICKLE-Report-D2_final.pdf
Vlieghe, J. (2016). Rapport D1: State-of-the-art betreffende mediagebruik bij jongeren en jongvolwassenen in België, Vlaanderen en Brussel - TICKLE-Report_D1.pdf. Retrieved 3 May 2017, from https://wise.vub.ac.be/tickle/wp-content/uploads/2015/09/TICKLE-Report_D1.pdf
Vocabulary Bank for Education. (2017). Retrieved 1 February 2018, from http://europeanschoolnet-vbe.lexaurus.net/vbe/browse
Watershed. (n.d.). Retrieved 6 April 2018, from https://www.watershedlrs.com/
Wax LRS. (2013). Retrieved 14 November 2017, from http://www.saltbox.com/wax-learning-record-store.html
Weibel, S. (1999). The state of the Dublin Core metadata initiative: April 1999. *Bulletin of the Association for Information Science and Technology*, *25*(5), 18–22.
White, R. W., Bailey, P., & Chen, L. (2009). Predicting user interests from contextual information. In: J. Allan, J. Aslam, M. Sanderson, C. Zhai and J. Zobel (Eds.)

Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval (pp. 363–370). ACM.

Wieringa, R. (2009). Design science as nested problem solving. In: V. Vaishanvi and S. Purao (Eds.) *Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology* (p. 1). Philadelphia, Pennsylvania: ACM Press.

Witkin, H. A., & Goodenough, D. R. (1980). Cognitive styles: essence and origins. Field dependence and field independence. *Psychological Issues*, (51), 1–141.

Wolf, C. (2003). iWeaver: towards' learning style'-based e-learning in computer science education. In: T. Greening and R. Lister (Eds.) *Proceedings of the fifth Australasian conference on Computing education-Volume 20* (pp. 273–279). Australian Computer Society, Inc.

Xerte. (2018). Retrieved 7 April 2018, from http://www.xerte.org.uk/index.php?lang=nl

Zervas, P., & Sampson, D. G. (2014a). Facilitating teachers' reuse of mobile assisted language learning resources using educational metadata. *IEEE Transactions on Learning Technologies*, *7*(1), 6–16.

Zervas, P., & Sampson, D. G. (2014b). The effect of users' tagging motivation on the enlargement of digital educational resources metadata. *Computers in Human Behavior*, *32*, 292–300.

A

APPENDIX A

Task description number : Task Doc 1 **Session date :**

Evaluator's name : Pascal **Session start time :**

Participant : **Session End Time :**

Actions involved in task description	User's remarks	Observer's comments
Click the button 'Student profiles'		
Click the button 'Add user'		
Complete all student profile details from the sheet in the interface and submit.		
Find the student profile and click the 'Edit' button		
Make the changes and save.		

Task description number : Task Doc 2 **Session date :**

Evaluator's name : Pascal **Session start time :**

Participant : **Session End Time :**

Actions involved in task description	User's remarks	Observer's comments
Finding and clicking the 'Create with Xerte' menu button.		
Follow the procedure to export the Xerte as a zip and save to the local pc.		
Click the 'Add Xerte object'.		
Fill in the metadata details from the sheet, select the local .zip and submit.		
Find the learning object and click the 'Edit' button		
Make the changes and save.		

Task description number : Task Doc 3 **Session date :**

Evaluator's name : Pascal **Session start time :**

Participant : **Session End Time :**

Actions involved in task description	User's remarks	Observer's comments
Finding and clicking the 'Create with H5P' menu button.		
Follow the procedure to copy the embed code to the clipboard.		
Click the 'Add H5P object'.		
Fill in the metadata details from the sheet, paste the embed code and submit.		
Find the learning object and click the 'Edit' button		
Make the changes and save.		

Task description number : Task Doc 4 **Session date :**

Evaluator's name : Pascal **Session start time :**

Participant : **Session End Time :**

Actions involved in task description	User's remarks	Observer's comments
Click the 'Learning objects manage' menu button.		
Find the learning object and click the 'Edit' button		
Use the *Tag explorer* to make the changes and save.		

Task description number : Task Doc 5 **Session date :**

Evaluator's name : Pascal **Session start time :**

Participant : **Session End Time :**

Actions involved in task description	User's remarks	Observer's comments
Click the 'Challenges manage' menu button.		
Find the 'Add' button and click it.		
Follow the 4 step process to create the challenge and submit.		

Task description number : Task Doc 6 **Session date :**

Evaluator's name : Pascal **Session start time :**

Participant : **Session End Time :**

Actions involved in task description	User's remarks	Observer's comments
Click the 'Learning objects manage' menu button.		
Sort the column 'title' in descending order.		
Choose the filter 'English' for the language and 'difficult' for the difficulty.		
Click 'Apply filters'.		
Find the search field above the table with learning objects.		
Fill in berrie and find the results in the table.		

Task description number : Task Doc 7 **Session date :**

Evaluator's name : Pascal **Session start time :**

Participant : **Session End Time :**

Actions involved in task description	User's remarks	Observer's comments
Click the 'Challenges by student' menu button.		
Select the student from the dropdown list and click the 'View' button.		
Find the answers for the statuses by hovering over the pie chart (not counting).		

Task description number : Task Doc 8 **Session date :**

Evaluator's name : Pascal **Session start time :**

Participant : **Session End Time :**

Actions involved in task description	User's remarks	Observer's comments
Click the 'Home' menu button.		
Find the last 10 actions on the right.		

B

APPENDIX B

System Usability Scale

	Strongly disagree 1	2	3	4	Strongly agree 5
1. I think that I would like to use this system frequently					
2. I found the system unnecessarily complex					
3. I thought the system was easy to use					
4. I think that I would need the support of a technical person to be able to use this system					
5. I found the various functions in this system were well integrated					
6. I thought there was too much inconsistency in this system					
7. I would imagine that most people would learn to use this system very quickly					
8. I found the system very cumbersome to use					
9. I felt very confident using the system					
10. I needed to learn a lot of things before I could get going with this system					

C

APPENDIX C

- **XTInitialize**

Here we initialize the xAPI connection to the LRS with the previously added function *initXAPI*. We also initialize all variables to be kept while tracking the student's activity and we sent a *launched* statement to the LRS to record that the activity has started. Note that pageresults is our variable that keeps track of each page in the learning activity whether it has been completed or not.

```
function XTInitialise()
{
//Initialize the xAPI user
initXAPI();

var objectToSend = {
        verb_id:"http://adlnet.gov/expapi/verbs/launched",
        target_id: MY_SERVER + "learning_objects_students/" + SCHEDULE_ID + "/index.htm",
        verb_display:"started",
        target_name:x_params["name"]
}

//INIT GLOBAL VARIABLES!!!
pageresults = [];
totalscore = 0;
numberOfExercises = 0;

sentXAPIstmt(objectToSend);
saveStatus(SCHEDULE_ID,"Started");
```

```
};
```

- **XTEnterPage**

Each time a page inside the learning activity is entered, this function will be executed. We will check if the page contains an interactivity exercise or game. If it does, the student needs to pass the exercise to get the page completed. For now, the exercises a student needs to pass are the ones in the first if. In the future, more types of interactivity exercises can be added.

```
function XTEnterPage(page_nr, page_name)
{
	my_current_page = x_pages[page_nr]["nodeName"];
if(my_current_page == "gapFill" || my_current_page == "grid" || my_current_page == "mcq" ||
my_current_page == "textMatch"|| my_current_page == "timeline" || my_current_page == "hangman"
|| my_current_page == "memory"){
		pagescoreSet = true;
	}
	else {
		pagescoreSet = false;
	}

	if(page_nr == x_pages.length - 1){
		pageresults[page_nr] = 'completed';
	}

	sentIfAllcompleted();

}
```

- **XTExitPage**

Each time a page of the learning object is left, this function will be executed.

```
function XTExitPage(page_nr)
{
	if(!pagescoreSet){
		pageresults[page_nr] = 'completed';
	}
	else{
		pagescoreSet = false;
	}

	sentIfAllcompleted();
}
```

- **XTSetPageScore**

The function that checks the page score.

```
function XTSetPageScore(page_nr, score)
{


if(score > 49) {
		numberOfExercises = numberOfExercises + 1;
		totalscore = totalscore + score;
}
else{
		alert("ATTENTION: you have not succesfully completed this page! Try again till
you score over 50% to complete this page.");
}

sentIfAllcompleted();

pagescoreSet = true;
}
```

- **sentIfAllCompleted**

This function will send the result to the LRS that was configured in *xapiconfig.js*. It will only sent the result if all pages of the challenge have been successfully completed.

```
function sentIfAllcompleted(){
       if(allCompleted(pageresults)){
                     var objectToSend = {
              verb_id:"http://adlnet.gov/expapi/verbs/completed",
              target_id:   MY_SERVER   +   "learning_objects_students/"   +   SCHEDULE_ID   +
"/index.htm",
              verb_display:"completed",
              target_name:"Categories exercise"
              };

              if(numberOfExercises == 0){
                     score_raw = 100;
                     score_scaled = 1;
              }
              else {
                     score_raw = totalscore/numberOfExercises;
                     score_scaled = (totalscore/numberOfExercises)/100;
              }

              objectToSend["result_score_max"] = 100;
              objectToSend["result_score_min"] = 0;
              objectToSend["result_score_raw"] = score_raw;
              objectToSend["result_score_scaled"] = score_scaled;
              objectToSend["result_succes"] = true;
              objectToSend["result_completion"] = true;


              sentXAPIstmt(objectToSend);
              alert("Congrats, you are finished!");
              saveStatus(SCHEDULE_ID,"Completed");
       }
}
```

D

APPENDIX D

1. **Services for Student management**

GET all the students and their details: http://188.166.127.224/root/index.php/studentREST1/students

GET all details of the student with ID 1: http://188.166.127.224/root/index.php/studentREST1/student/id/1

DELETE the student with ID 2: http://188.166.127.224//root/index.php/studentREST1/student/id/2

2. **Services for Learning Objects management**

GET all learning objects and their details: http://188.166.127.224/root/index.php/LearningObjectREST1/learningobjects

GET all details of the learning object with ID 73: http://188.166.127.224/root/index.php/LearningObjectREST1/learningobject/id/3

GET a sorted list of the learning objects that have best matches with the interests of student with ID 1 (most relevant first): http://188.166.127.224/root/index.php/LearningObjectREST1/findmatchesfor/studentid/1

GET the ILOX-LOM xml representation of the learning object with ID 2: http://188.166.127.224/root/index.php/learningObjectREST1/ILOXLOMlearningobject/id/2/format/xml

3. **Services for Challenge management**

GET all details of the challenge with ID 65 (as stored in table): http://188.166.127.224/root/index.php/scheduleREST1/schedule/id/65

GET the details of all available challenges:

http://188.166.127.224/root/index.php/scheduleREST1/schedules/

POST the learning object with ID 73 (LOlangID = 73) as a new challenge for student with ID 1 (studID = 1). You will receive the URL for the student to complete as result:

http://188.166.127.224/root/index.php/scheduleREST1/schedule/

DELETE the challenge with ID 68:

http://188.166.127.224/root/index.php/scheduleREST1/schedule/id/68

GET full details of the challenge with ID 66 with all foreign keys made explicit, most comprehensive challenge details available: http://188.166.127.224/root/index.php/scheduleREST1/schedulereadable/id/66

GET the status ('Open', 'Started, but not finished' or 'Completed') of the challenge with ID 64:

http://188.166.127.224/root/index.php/scheduleREST1/checkLRSstatus/scheduleID/64

4. **Services for tags**

GET all tags that are being used by at least one learning object in the database:

http://188.166.127.224/root/index.php/LearningObjectREST1/allTagsUsedByLearningObjects

GET all the learning objects in the database with tag id 3528:

http://188.166.127.224/root/index.php/LearningObjectREST1/allLearningObjectsWithTag/tagid/3528

GET all tags that are within the interest of at least one student in the database:

http://188.166.127.224/root/index.php/studentREST1/allTagsUsedByStudents

GET all students that are interested in the tag with ID 1482:

http://188.166.127.224/root/index.php/studentREST1/allStudentsWithInterestTag/tagid/1482

5. Services for keywords

GET all keywords that are being used by at least one learning object in the database:

http://188.166.127.224/root/index.php/LearningObjectREST1/allKeywordsUsedByLearningObjects

GET all the learning objects in the database with keyword *fruit*: http://188.166.127.224/root/index.php/LearningObjectREST1/allLearningObjectsWithKeyword/keyword/fruit

GET all keywords that are within the interest of at least one student in the database: http://188.166.127.224/root/index.php/studentREST1/allKeywordsUsedByStudents

All data returned by the REST API is in the form of JSON. For illustration purposes, we include here an example of part of JSON response for a challenge request.


```
{
        "schedule_id": 64,
        "added_by_teacher_id": 4,
        "student_id": 1,
        "learning_objects_language_id": 1,
        "date": "2018-03-04",
        "url": "http://188.166.127.224/root/learning_objects_students/64/index.htm",
        "status": "Started, but not completed",
        "last_checked": "2018-04-05T07:06:10+0000",
        "deleted": "no",
        "learning_object_details": {
            "id": 78,
            "learning_resource_type": "educational game",
            "interactivity_level": "low",
            "intended_end_user_role": "learner",
            "context": "compulsory education",
            "typical_age_range": "14-16",
            "difficulty": "1",
            "cost": "no",
            "copyright": "no",
            "classification_purpose": "discipline",
            "classification_source": "(\"x-none\",\"LRE-0001\")",
            "educational_level": "1",
            "education_year": "3",
            "lo_type": "combined",
            "paedagogic_goal": "2",
            "deleted": "no",
            "learning_objects1_contributions_id": 133,
            "learning_objects_id": 78,
            "user_id": 4,
            "role": "author",
            "date": "2018-03-04",
            "learning_objects1_language_id": 1,
            "learning_objects1_id": 78,
            "language": "en-GB",
            "catalogue": "http://188.166.127.224/root//learning_objects_base/",
            "entry": "1",
            "title": "Berry memory",
            "description": "",
            "language_level": "1",
            "sensitive_remarks": "",
            "created_with": "<iframe src=\"http://188.166.127.224/wp-admin/admin-
ajax.php?action=h5p_embed&id=5\" width=\"958\" height=\"211\" frameborder=\"0\"
allowfullscreen=\"allowfullscreen\"></iframe><script src=\"http://188.166.127.224/wp-
content/plugins/h5p/h5p-php-library/js/h5p-resizer.js\" charset=\"UTF-8\"></script>",
            "original": "yes",
            "learning_objects1_language_deleted": "no",
```

```
            "language_level_id": 1,
            "language_level_level": "easy",
            "difficulty_id": 1,
            "difficulty_level": "easy",
            "education_level_id": 1,
            "education_level": "General Education Level",
            "paedagogic_goal_level_id": 2,
            "paedagogic_goal_level": "Understanding",
            "paedagogic_goal_level_description": "The student can explain ideas or concepts: classify, describe, discuss, explain, identify, locate, recognize, report, select, translate",
            "all_keywords": [
                "fruit"
            ],
            "taxonomy_tags": [
                {
                    "ID": 3528,
                    "word": "biological sciences"
                }
            ],
            "sensitives": [
                "politics",
                "fruit laws"
            ],
            "all_thumbnails": [
                "http://188.166.127.224/root/learning_objects_base/1/media/thumbs/0.jpg",
                "http://188.166.127.224/root/learning_objects_base/1/media/thumbs/thumb0.jpg",
                "http://188.166.127.224/root/learning_objects_base/1/media/thumbs/1.jpg",
                "http://188.166.127.224/root/learning_objects_base/1/media/thumbs/thumb1.jpg",
                "http://188.166.127.224/root/learning_objects_base/1/media/thumbs/2.jpg",
                "http://188.166.127.224/root/learning_objects_base/1/media/thumbs/thumb2.jpg",
                "http://188.166.127.224/root/learning_objects_base/1/media/thumbs/3.jpg",
                "http://188.166.127.224/root/learning_objects_base/1/media/thumbs/thumb3.jpg"
            ]
        },
        "student_details": {
            "user_id": 1,
            "first_name": "Jansegers",
            "last_name": "Reinhard",
            "day_of_birth": 1,
            "month_of_birth": 1,
            "year_of_birth": 1999,
            "email": "pascal_pieters@yahoo.com",
            "gender": "Male",
            "ethnicity": "",
            "monthly_income": "",
            "diploma_mother": "",
            "diploma_father": "",
            "occupation_mother": "",
            "occupation_father": "",
            "parent_school_involvement": null,
            "parent_expectations": null,
            "parents_notes": "",
            "other_family_members": "",
            "teacher_relations": "",
            "peers_school": "",
            "peers_others": "",
            "peers_siblings": "",
            "neighborhood": "",
            "nationality": "French",
            "nationality_father": "French",
            "nationality_mother": "French",
            "language_mother": "nl",
            "language_father": "fr",
            "language_siblings": "fr",
            "language_friends": "fr",
            "writing_direction": "Left To Right",
            "sensitive_content": "",
            "smartphone_access": "Yes, but shared with others",
            "computer_access": "Yes, I have my own device for personal use",
            "internet_access": null,
            "internet_access_wifi": "Wifi",
            "internet_access_data_subscription": "Data Subscription",
            "internet_access_homespot": "Homespot",
            "smartphone_use": "Couple of times a week",
            "smartphone_lookup": "Weekly",
            "smartphone_email": "Weekly",
            "smartphone_sms": "Couple of times a week",
```

```
        "smartphone_FBMessenger": "Weekly",
        "smartphone_snapchat": "Couple of times a week",
        "smartphone_instagram": "Weekly",
        "smartphone_youtube": "Couple of times a week",
        "smartphone_facebook": "Weekly",
        "school_attendance": null,
        "good_behavior": null,
        "school_activities": null,
        "expectations": null,
        "goodsubjects": "",
        "badsubjects": "",
        "homework": null,
        "motivation_school": null,
        "autonomy": null,
        "competence": null,
        "witkin": null,
        "VARK": null,
        "years_completed": "",
        "diplomas_attained": "",
        "schoolmobility": "",
        "dropouthistory": "",
        "facebook_username": "",
        "instagram_username": "",
        "snapchat_username": "",
        "telnumber": "",
        "xapi_username": "scalli123",
        "xapi_password": "nilvolentibusarduum8",
        "xapi_mbox": "ppieters@vub.ac.be",
        "deleted": "no",
        "education_language": "nl",
        "education_language_level": "1",
        "smartphone_music": "Couple of times a week",
        "smartphone_video": "Weekly",
        "smartphone_games": null,
        "language_level_id": 1,
        "language_level_level": "easy",
        "user_core_sensitives": [],
        "user_added_sensitives": [],
        "all_keywords": [],
        "taxonomy_tags": [
            {
                "ID": 33075,
                "word": "urban development"
            },
            {
                "ID": 21264,
                "word": "music"
            }
        ]
    }
}
```