

Article

TESMA: Requirements and Design of a Tool for Educational Programs [†]

Nicolas Guelfi ^{*‡}, Benjamin Jahić [‡] and Benoît Ries [‡]

Université du Luxembourg, 2, avenue de l'Université, L-4365 Esch-sur-Alzette, Luxembourg;
benjamin.jahic@uni.lu (B.J.); benoit.ries@uni.lu (B.R.)

* Correspondence: nicolas.guelfi@uni.lu; Tel.: +352-466-644-5251

† This paper is an extended version of our paper published in Nicolas Guelfi, Benjamin Jahić and Benoît Ries, TESMA: Towards the Development of a Tool for Specification, Management and Assessment of Teaching Programs, published in the Proceedings of the 2nd International Conference on Applications in Information Technology (ICAIT-2016).

‡ These authors contributed equally to this work.

Academic Editors: Evgeny Pyshkin, Vitaly Klyuev and Alexander Vazhenin

Received: 2 December 2016; Accepted: 18 March 2017; Published: 22 March 2017

Abstract: Defining and managing teaching programs at universities or other institutions is a complex task for which there is not much support in terms of methods and tools. This task becomes even more critical when the time comes to obtain certifications w.r.t. official standards. In this paper, we present an on-going project called TESMA, whose objective is to provide an open-source tool dedicated to the specification and management (including certification) of teaching programs. An in-depth market analysis regarding related tools and conceptual frameworks of the project is presented. This tool has been engineered using a development method called Messir for its requirements elicitation and introduces a domain-specific language dedicated to the teaching domain. This paper presents the current status of this project and the future activities planned.

Keywords: educational programs development; tool-support; software engineering; domain-specific languages; automatic generation

1. Introduction

The University of Luxembourg [1] is a young university (created in 2003). The Grand Duchy of Luxembourg is a multicultural, multilingual country in Western Europe located at the intersection of France, Germany and Belgium. Its only public university was founded in 2003 and embodies its multicultural spirit. In 2016, 860 professional experts from 20 different countries were working in different departments and research areas. In this multicultural “start-up” context, we have been setting up new programs at bachelor, master and doctorate levels providing different education certificates. All of those programs are offered to 6172 students from 115 different countries by our three faculties. The need for a tool to support the task to define and manage (including certification) the education program came rapidly. The market analysis for this category of tools showed that no tool was available. A project has been started to engineer a method and a tool to support those needs. This project has been conducted following a software engineering process that had the following main steps:

- Requirements analysis: to provide the initial requirements for the TESMA tool, a requirement specification document has been produced using the Messir method [2].
- Design: to state the main choices concerning the TESMA architecture and interfaces.
- Implementation: to reach an operational system usable for validation w.r.t. the requirements.
- Those steps have been performed iteratively to produce the TESMA tool in an incremental way.

The content of this article provides details on the requirements analysis and design of the TESMA tools. This article extends our conference paper [3] by providing the following additional content: two extensive surveys exploring existing tool features and the conceptual frameworks used by a number of educational institutions carefully selected over the world; and based on the surveys' results, the presentation of a new architectural component, the Moodle page generator, in Section 2.3.4.

2. Results

2.1. Surveys

We have performed two surveys in the context of the TESMA project. On the one hand, a large number of tools related to the handling of educational programs have been surveyed; on the other hand, educational conceptual frameworks of international institutions have been explored. Section 4 describes the methods and materials that we used to produce the results of the surveys presented in this section.

2.1.1. Survey on Educational Programs Management Tools

This survey has been performed in two steps: a primary study and a secondary study [4] (in this article, we are using some of the terminology introduced by Kitchenham et al. [4] applied to the context of our article, i.e., our technological watchdog survey and our conceptual frameworks survey). The main objective of this tool survey is, taking as input the set of TESMA requirements that we elicited, to search for existing tools such that we are able to choose between:

1. Extending an existing tool or
2. Implementing a new ad hoc tool

The primary study explored the existing tools, related to our required feature categories. This primary study resulted in a collection of around 200 different tools related to educational program management. Then, based on this large set of tools, we have performed a secondary study, which resulted in a reduced set of 27 tools, listed in Appendix A.

During the TESMA requirement elicitation phase performed prior to this survey, we had identified 373 different features grouped in 101 different categories that define the required feature categories for our TESMA tool to be implemented. The names of these feature categories are listed in Appendix C. During the secondary study, for each of the 27 tools, we have identified the operations provided by the tool based on its documentation and on its publicly available documentation. Each identified tool operation has been mapped to one of the TESMA feature category that we defined.

Table 1 presents the feature categories coverage for the six most interesting tools. We can note that the closest tool to TESMA in terms of feature categories coverage is Moodle. (Note that in terms of features coverage Blackboard [5] is closely related to Moodle, but we could not survey it because of not being able to access to a demo license. Anyhow, this tool is not open-source, thus would not have been compatible with open-source orientation of TESMA.) Moodle is an open-source learning management system that supports the description of programs and courses; and the sharing of information and course material between teachers and students.

With respect to our study objective, as a result of our secondary study, we conclude that there is not any tool covering enough of the TESMA feature categories. The best tool being Moodle with a coverage of our feature categories of around 50%. Thus, we have decided to:

1. implement an ad hoc tool, providing the feature categories not covered by Moodle, e.g., specification of educational programs with a DSL, and
2. implement a Moodle page generator component in our tool that creates Moodle-compliant web pages, presented in Section 2.3.4. The objectives of this new component are two-fold:

- (a) benefit from the existing web front-end of Moodle.
- (b) ease the adoption of our tool by Moodle users.

Table 1. The six most interesting tools resulting from our study and their coverage w.r.t. the TESMA features categories.

Tools	TESMA Features Categories		
	Covered	Not Covered	Coverage
Moodle	52	49	52%
Fedena	43	58	43%
Sakai Project	34	67	34%
Schooltool	22	79	22%
Canvas	16	85	16%
D2L	15	86	15%

2.1.2. Survey on Educational Programs Concepts

Based on the TESMA tool's requirements analysis specification, presented in Section 2.2, we performed a conceptual framework survey. This survey consists in studying the concepts used in existing educational programs of carefully selected universities in the world.

The main objective of this conceptual frameworks survey is to validate the set of our TESMA domain concepts, presented briefly in Section 2.2, by:

1. Evaluating if the TESMA domain concepts cover all the concepts of the selected universities. This would answer a question like: does the current TESMA concept model contain enough concepts to specify all the educational programs needs of the selected universities?
2. Evaluating the coverage of the existing universities's concepts against the full set of TESMA domain concepts. This evaluation would give a quantitative hint on how much more concepts could be covered by the selected universities if they would use our TESMA tool.

This study has been performed on a set of 10 universities listed, together with their websites and programs, in Appendix B. During the TESMA requirements analysis phase, we had identified 85 domain concepts. The survey of education programs concepts consisted in the analysis of the concepts of the selected institutions and compared them with the 85 TESMA domain concepts. Our study showed us that the surveyed institutions are often using different naming conventions, except for general concepts like courses, elective courses, programs, etc., even though, all universities' concepts might differ in their naming conventions or internal structure, their semantics are often compatible.

Table 2 presents the results of the universities conceptual frameworks study. The table shows the selected universities, with the number of concepts that we identified for each institution, the number of institution concepts that match with TESMA concepts (or not). The covered institutions concepts are concepts that we considered to be compatible in terms of their informal semantics with TESMA concepts. We also display, for information, in the last column of the table, the proportion of the university concepts in terms of the overall TESMA concepts defined.

With respect to our study objectives, this conceptual frameworks study reveals that:

1. most of the universities concepts are covered in TESMA, at a proportion of 70% or more. Thus, it is possible for universities to use the TESMA conceptual framework to specify most of their concepts.
2. the coverage of TESMA concepts by all institutions is rather low, below 50%. This is partly due to the low number of universities having a course certification policy. Thus, most universities would gain using our domain model, it would improve the number of concepts used in their educational programs specifications.

In order to attract a larger set of institutions, a generic tool needs to be developed. This tool needs to be adaptable to the universities concepts and naming conventions. Underlying to this tool's requirements, the first version of the TESMA concept model will need to be improved. These constraints will improve the compatibility of the tool at other institutions and increase the institutions concepts coverage to 100%.

Table 2. Results of the survey on the 10 universities' conceptual frameworks.

Universities	Institutions Concepts			TESMA Concepts	
	Covered	Not Covered	Total	Coverage	Coverage
University of McGill	37	0	37	100%	44%
The Hong Kong University of Sci. and Tech.	17	0	17	100%	20%
University of Geneva	14	0	14	100%	17%
King Abdulaziz University	17	1	18	95%	20%
University of Luxembourg	50	5	55	91%	59%
Innopolis University	14	2	16	88%	17%
University of Southern California	24	4	28	86%	29%
University de Rennes 1	9	3	12	75%	11%
National University of Singapore	18	7	25	72%	22%
The University of New South Wales	12	5	17	71%	15%

2.2. TESMA Requirements

Following part of the Messir method [2], presented shortly in Section 4.2, we have elicited the actors that are concerned by teaching programs. They are:

- The institution director who represents the institution and validates new programs, courses and course modifications (e.g., the dean of a faculty, the head of a teaching unit, etc.).
- The program director who specifies his programs and validates course modifications made by instructors.
- The instructor, who specifies, manages and maintains the courses he gives.
- The student, who tunes his curriculum (elective courses, etc.) and receives information about his curriculum.
- The secretary, who is a delegate of any of the institutional actors (institution director, program director or instructor) and also ensures interoperability with other institution information systems.
- The quality officer who evaluates and validates the programs with respect to the universities internal laws. He's also responsible of program certification processes.

You can find in Figure 1, a use-case model made in the context of the Messir method that displays the actors contributing to the high-level summary use-case dedicated to managing a teaching program.

The concepts managed by the TESMA actors are analyzed and specified in the Messir concept model which is an UML [6] class diagram. Concept models diagrams are partial views on the domain concepts handled by the system under study. The concept model can be seen as a result of domain analysis. Among all the concepts that are necessary to specify the operations executed by the actors we have:

- concepts related to the actors and for which TESMA has to handle an internal representation: students, instructors, etc.
- concepts related to the programs: course details, teaching periods, course evaluation, etc.
- concepts related to program certification: standard description, standard coverage by an existing program, etc.

In Messir those concepts are specified using an UML [6] class diagram, Figure 2 provides a partial view on some of the concepts related to a program (institution, program, courses).

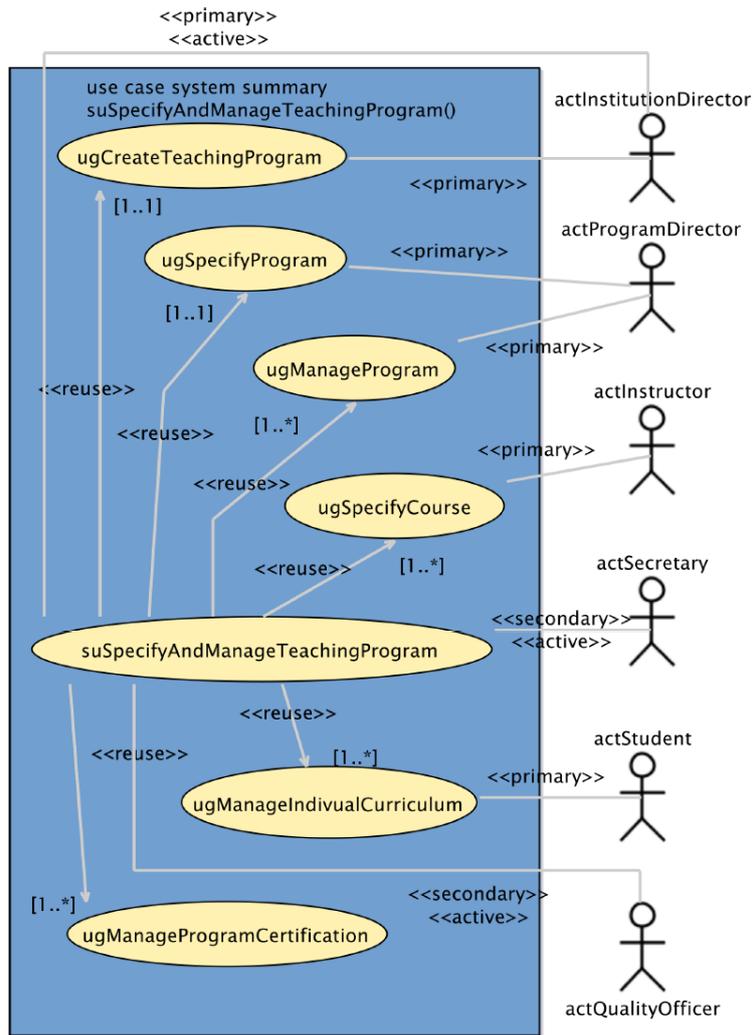


Figure 1. TESMA summary use-case model.

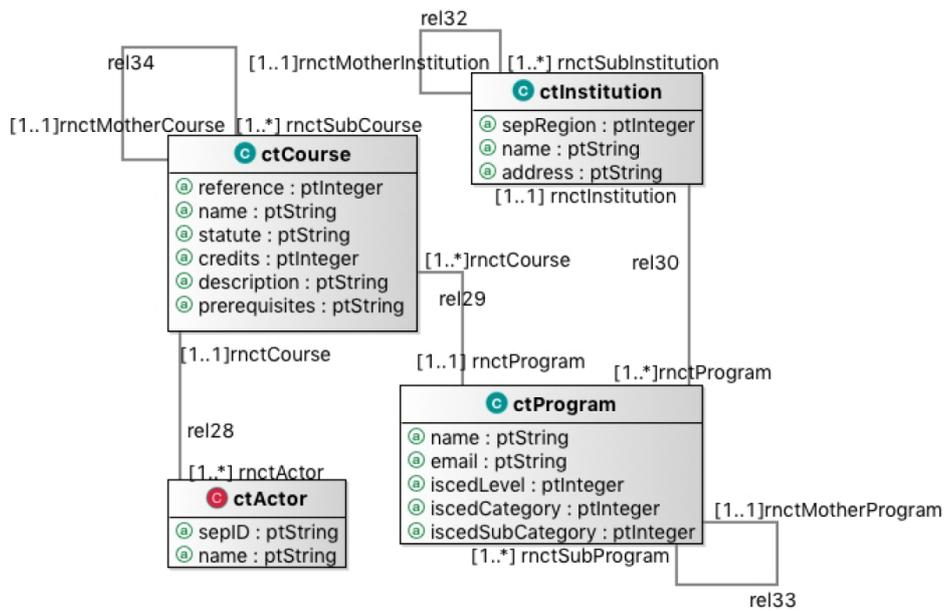


Figure 2. TESMA concepts.

The requirements analysis phase has allowed to determine a first version of the functionalities and data that should be handled in the first increment. The next section presents the design of this first version.

2.3. TESMA Tool Set Development

After having analyzed the TESMA actors, concepts and functionalities, we have started to design a first version of the TESMA tool set. A major design choice made is to allow the specification of programs using a Domain-Specific Language (DSL) [7] defined using the Xtext [8] framework. Xtext is an open-source framework for the development of textual domain-specific languages and the creation of textual editors. Thus, we have designed TESMA as an Eclipse plugin, i.e., a java program extending the Eclipse workbench [9]. Eclipse is an open-source extensible development workbench that benefits from a large community of developers. TESMA is composed of four main architectural components illustrated in Figure 3 at the top of the diagram. Our components are based on stable Eclipse plugins themselves based on the Eclipse Modeling Framework (EMF) [10]. EMF is an open-source modeling framework part of the Eclipse workbench providing tools to manage structured data models. EMF is used as the underlying core library handling the TESMA model.

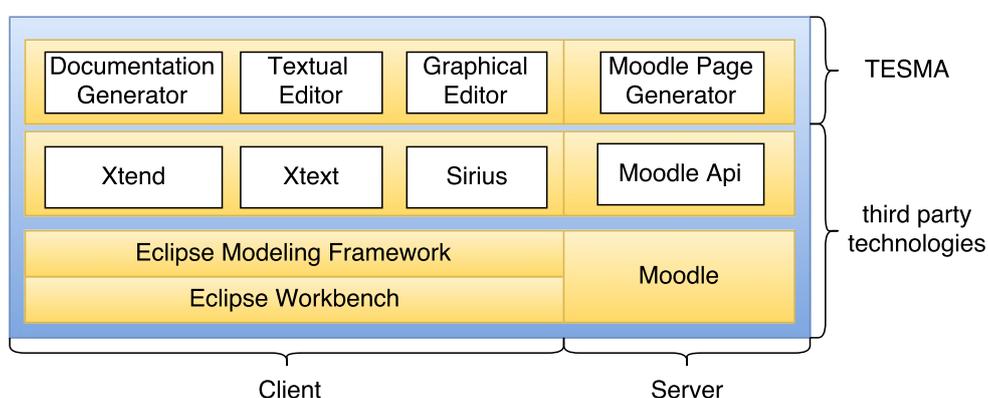


Figure 3. Architectural components overview.

In the following sections, we present the different component of the TESMA plugin: textual editor, graphical editor, documentation generation, Moodle page generator.

2.3.1. Textual Editor

This section presents the features of the Textual Editor; and the design, implementation and test cases of the grammar of the TESMA DSL [11] used for specifying the teaching programs and courses.

The main feature of the Textual Editor is to allow the specification of the teaching programs (this specification is called TESMA model in the remaining part of this paper) with the TESMA DSL. It also offers other supporting features, as for instance: syntactical validation rules, syntax highlighting, templates proposal, auto-formatting, scoping, quick fixes and auto-completion, hovers, outline view, compare view, code folding, etc. The aim of these features is to support the specifier during the specification of the TESMA model. These features improve the readability, understanding, accessibility and validation of the specification. The time to understand and to learn how to use the syntax is reduced by using proposals, auto-formatting and auto-completion. Thanks to the outline view, the user has a graphical overview of his/her specified TESMA models. Parts of the specification can be hidden using code folding to improve the readability. The syntactical validation rules and the auto-completion help the user of the DSL to improve the learnability of the syntax and specify effectively a valid TESMA Model. The result from concept model analysis, partly shown in Figure 2, has been taken as input for the design of the DSL and in particular for the specification of its grammar in Xtext.

The TESMA DSL is designed to be intuitive, customizable and loosely-coupled. In order to have an intuitive DSL, we have chosen to design its grammar using mainly keywords in natural language. Institutions often use different terminologies for the concepts used in our approach, this is why we designed the grammar of our DSL to be customizable. The institutions have the possibility to choose their own naming conventions. Our design should allow non-computer experts to use the TESMA DSL as it uses natural language keywords, no logical operators and control flow statements. The rules of the grammar are loosely coupled, i.e., optional cross-references are mostly used instead of containment relations. Lastly, thank to loosely coupled grammar rules, TESMA supports multiple user specifications. The TESMA model can be specified independently by different persons. Thus, the specifications can be merged together afterwards.

The TESMA DSL has been implemented using the Xtext framework. During the DSL design, we mapped the identified concepts to Xtext grammar rules. The grammar represents the metamodel of TESMA models. It has been developed in the .Xtext file of the parent Xtext project and consists of a set of rules that represent the syntax of the TESMA DSL. The following types of rules are used to develop the grammar: fixed keywords, user-dependent keywords, consecutive words, optional words or complete phrases, different types of relations, etc. Figure 4 shows a part of the TESMA grammar that illustrates the fundamental TESMA concepts (institutions, programs, courses). Once the grammar has been defined, an EMF Ecore model and a generator model (genmodel) has been generated using the Xtext artifact generator. The EMF Ecore model contains basic information about the defined grammar, e.g., classes, attributes and relations. The genmodel contains some additional properties for the code generation that are not part of the metamodel itself, e.g., generated source code, path and file information. A part of the generated source code is the model validation code, the scoping and a generator class. Thanks to the code generation, the validation of the TESMA model has been developed inside the TESMA validator. A set of rules has been defined to help the specifier to correctly specify TESMA models. The scoping allows a user defined content assistant and linking. Thanks to the generator, we have generated plenty of documents, see Section 2.3.3.

```

17= Model:
18 elements+=CtInstitution | ctProgram | ctCourse |
19 ctStandard | ctActor | ctTask | ctTest |
20 ctCategory | ctCriteria)*
21 ;
22
23# * Institution => Faculties, Universities, SubUniversities etc.[]
24
25@ ctInstitution:
26 (hide=HIDDEN)? 'Institution' name=ID ('partOf' rncctMotherInstitution=[ctInstitution|FQN])? ('con:
27 ('name' label=STRING)
28 ('region' sepRegion=INT)
29 ('address' address=STRING)
30 ('programs' programs=[ctProgram|ID](',' programs+=[ctProgram|FQN])*)?
31 ('description' description=STRING)?
32 ('currentSituation' currentSituation=STRING)?
33 ('gastronomie' gastronomie=STRING)?
34 ('contact' contact=STRING)?
35 '};
36
37# * Program given in a institution[]
38
39@ ctProgram :
40 (hide=HIDDEN)? 'Program' name=ID ('partOf' rncctMotherProgram=[ctProgram|FQN])?('contains' rncctSul
41 ('name' label=STRING)
42 ('iscsd' iscedLevel=INT (' iscedCategory=INT (' iscedSubcategory=INT)??)?)
43 ('description' description=STRING)
44 ('programdirector' rncctProgramdirector=[ctInstructor|FQN])
45 ('prerequisites' prerequisites=STRING)?
46 ('requisites' requisites=STRING)?
47 ('costs' cost=STRING)?
48 ('language' languages+=LANGUAGE(',' languages+=LANGUAGE)*)
49 ('email' email=STRING)
50 ('weblink' weblink=STRING)
51 ('courses' courses+=[ctCourse|FQN](',' courses+=[ctCourse|FQN])*)?
52 (terms+=ctTerm)*
53 (modules+=ctModule)*
54 '};
55
56# * Definition of a course []
57
58@ ctCourse:
59 (hide=HIDDEN)? 'Course' (nature=NATUREOFCOURSE) name=ID ((programtype='belongs' | programtype='ir
60 ('name' label=STRING)
61 ('reference' reference=REFERENCE)
62 ('corecourse' corecourse=[ctCourse|FQN])?
63 ('academicyear' beginyear=INT//endyear=INT)

```

Figure 4. TESMA grammar.

In our tooling context, we present two possible ways to test the TESMA DSL. One possible choice is the unit testing feature of Xtext. Using this feature, we are able to implement automated test that check if a piece of the source code is correct for some particular case. The unit testing feature uses JUnit [12] as unit testing framework for Java applications. The aim of these unit testing features is to improve the maintainability and quality of the software product. Xtext supports such testing on the textual editor or on the metamodel itself. A second possibility is to test the TESMA DSL using some concrete examples. Our test cases are based on the University of Luxembourg with its three faculties and its various teaching programs that propose courses in plenty of different domains. To test the features, the validation and the usability of our grammar, we have specified one course given in a program of the University of Luxembourg, see Figure 6. These four development steps, Design, Features, Implementation and Test are executed iteratively.

2.3.2. Graphical Editor

The Graphical Editor provides a representation of the TESMA model in a tabular view and offers the possibility to modify the TESMA model. The graphical editor provides typical table handling features like data sort, import/export from/to Excel sheets, hide/show columns, multiple rows selections.

The technology used to develop our graphical editor is Sirius [13], an open-source software Eclipse project that eases the creation of custom graphical modeling workbenches. Both Xtext and Sirius are based on EMF, which allows the TESMA tool-support to interact between Xtext and Sirius using EMF as underlying-core library for the TESMA model as represented in Figure 5.

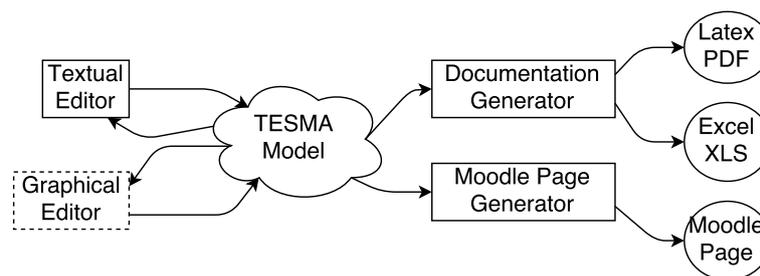


Figure 5. TESMA file generation.

Thanks to our tabular format, the graphical editor is intuitive and usable by non-computer experts. All the program's attributes are easy to access and modify. The modifications can be performed directly inside the graphical editor view.

2.3.3. Documentation Generator

The main feature of the documentation generator is to generate documents of different types, like Excel sheets, CSV files and PDF files. The documentation generator may be configured to produce a customized PDF file, e.g., by not generating some of the sections inside the pdf files.

The technologies used to develop the documentation generator are Xtend [8], Latex and the apache.poi library, for handling Excel sheets. Xtend is a programming language based on Java. It provides a compact syntax and eases the generation of natural language text. Latex is a document preparation system, which uses libraries, keywords and plaintext for writing scientific documents in pdf format. Finally, the apache.poi library provides the necessary tools for generating Excel sheets, which are used as teaching material.

The documentation generator has been designed to ease information retrieval in the generated Latex files. Additionally, it is designed to automatically update the final report, when the user manually adds data into the reserved appropriate folders. Finally, the different Latex files are imported inside one Latex file, which is compiled into a pdf file containing the program description.

2.3.4. Moodle Page Generator

The main feature of the Moodle page generator is to generate Moodle web pages displaying the content of the courses specified as a TESMA textual specification.

The current design is to use the Moodle APIs to create as much information as possible in the Moodle site using the web services provided. The technologies used to develop the moodle page generator are: on the one hand, REST web services offered by Moodle called using HTTP protocol in order to create a new page or a new user account in the TESMA Moodle server; on the other hand, execution of SQL queries modifying directly the Moodle server's SQL database, for the features that are not yet implemented as REST web services.

Current experimentation implementation of this component support one-way generation, i.e., taking as input TESMA specification of courses and configuring a Moodle website to display the courses specified.

2.4. Illustration

We illustrate the TESMA approach with a course of a Master program named "Software Engineering Environment" (SEE) at the University of Luxembourg. Figure 6 is a screenshot of the TESMA tool-support in the Eclipse environment.

```

Course required MICS233 belongs MICS {
  name "Software Engineering Environments"
  reference 2.33
  academicyear 2016/2017
  term MICS.Semester3
  module MICS.Semester3.module43
  hoursPerWeek 2
  totalHours 120
  description "Software engineers need means for quality engineering ..."
  credits 5
  languages "french", "english"
  weblink "https://www.uni.lu/MICS/SEE"
  coursemoderator guni
  //Teaching and Students
  boards SEEBoard
  promotions MICS_SEE_CLASS
  organisation org1 typeof lecture {
    instructor guni : hours 30,weight 1,language "english, french"
    instructor beri : hours 30,weight 1,language "english, french"
    instructor alca : hours 30,weight 1,language "english, french"
  }
  Period (Semester,0,0) start 17.09.2015 end 17.12.2015 {
    //LECTURE PERIOD 1
    Period (LecturePeriod,1,0) start 17.09.2015 end 8.10.2015{
      Period (Lecture1,1,1) start 15.10.2015 from 10:00 to 11:30{tasks t1a,t
      Period (Lecture2,1,2) start 15.10.2015 from 10:00 to 11:30{tasks t2a,t
      Period (Lecture3,1,3) start 15.10.2015 from 10:00 to 11:30{tasks t3a,t
      Period (Lecture4,1,4) start 15.10.2015 from 10:00 to 11:30{tasks t4a,t
    }
    Period (Checkpoint1,2,0) start 15.10.2015 from 10:00 to 11:30{tests oralCh
    Period (FinalExam,3,1) start 15.01.2016{ tests finalExam}
  }
}

```

Figure 6. An example of course specification in the TESMA textual editor.

The TESMA model describing the SEE course has been specified using the approach described in this article. The teaching program description of this example includes a number of textual files using the TESMA DSL syntax. The course run information is illustrated in Figure 6 by specifying the course teaching team organization and dividing the teaching term into small periods and defining the tasks, tests for each period. The tasks and tests are defined in separate folders and referenced to the teaching period. At that point, TESMA is able to generate a part of a Moodle page, the Teaching Material, like evaluation criteria, task lists and course information. Figures 7 and 8 are screenshots of the generated moodle page based on the SEE course described with the TESMA specification resp.

screenshots of the course syllabus of the SEE course that is included inside the generated TESMA report pdf file.

All other concepts can be specified using the TESMA DSL including the coverage of an education standard by an education program.

In this case study for the SEE course, we created for each category of TESMA model element a file containing all information related. In this case, one institution, one program and one course have been specified, which represent about 10 textual files. We defined 10 instructors and seven students for this example case, which are grouped in a single file. The total description in our case needs about 500 lines of specification text (>1000 in case of certification). The specification text size mostly depends on the preciseness of the specifier. If the specification is done in details, the number of lines increases quickly. In general, it could vary from 100 to 1500 lines.

NAVIGATION

- Dashboard
 - Site home
 - Site pages
- Current course
 - MICS233_**
 - Participants
 - Badges
 - 1.1. Lecture1
 - 1.2. Lecture2
 - 1.3. Lecture3
 - 1.4. Lecture4
 - 2. CheckPoint1
 - 3.1. ProjectSession1a
 - 3.2. ProjectSession1b
 - 3.3. ProjectSession1c
 - 3.4. ProjectSession1d
 - 4. CheckPoint2
 - 5.1. ProjectSession2a
 - 5.2. ProjectSession2b
 - 5.3. ProjectSession2c
 - 6. CheckPoint3
 - 7.1. OralExam
 - 7.2. CompleteProject
 - 7.3. Paper
- Courses

ADMINISTRATION

- Course administration
 - Turn editing on
 - Edit settings
 - Users
 - Filters
 - Reports
 - Grades

Full Course Description

Software engineers need means for quality engineering of the provide to customers. There exists three main categories of m belonging to the development life cycle, workbenches which c tools to cover a subpart of the software life cycle, and environi workbenches in order to cover the full life cycle. The first part i existing tools categories and describes the main technologies more attention is given to the Eclipse framework, rich textual e Xtext, advanced graphical editing using Sirius. Low level techr EMF, SVN, Jira. In the second part of this lecture, the students definition and design of a software engineering environment tc engineering project given as input.

Academic Year: 2016/2017
Term: Semester3
Module: module43
Hours per Week: 2
Total Hours: 120
Credits: 5
Languages: french, english
Course Moderator: Nicolas GUELF
Teacher: Nicolas GUELF
Teacher: Benoit Ries
Teacher: Alfredo Capozzucca
Course external webpage: <https://www.uni.lu/MICS/SEE>

1.1. Lecture1

Description:
[Description of Lecture1 is currently empty]

Date: 15.10.2015 **from** 10:00 **to** 11:30
Consists of: tasks t1a, t1b

1.2. Lecture2

Description:
[Description of Lecture2 is currently empty]

Date: 15.10.2015 **from** 10:00 **to** 11:30
Consists of: tasks t2a, t2b

Figure 7. Generated course in Moodle.

Chapter 3

Courses

3.1 MICS - Software Engineering Environments

3.1.1 Software Engineering Environments - 2.33

Core Course Information

Description Software engineers need means for quality engineering of the software and systems they contribute to provide to customers. There exists three main categories of means: tools for supporting any unit of task belonging to the development life cycle, workbenches which combine in an integrated way two or more tools to cover a subpart of the software life cycle, and environments which combine tools and workbenches in order to cover the full life cycle. The first part of this lecture provides a panorama of the existing tools categories and describes the main technologies on which those tools are built. Among them more attention is given to the Eclipse framework, rich textual editing for domain specific languages using Xtext, advanced graphical editing using Sirius. Low level technologies used in this lecture include: Java, EMF, SVN, Jira. In the second part of this lecture, the students work during practical sessions on the definition and design of a software engineering environment to be set up in the context of a concrete engineering project given as input.

Table 3.1: Course Information

Software Engineering Environments - 2.33	
Academic Year	2016/2017
Core Program	
Name	Master in Information and Computer Science
Term	3.0
Module	3.20
Credits	5
Hourse/Week	2
Total Hours	120
Languages	11
Weblink	https://www.uni.lu/MICS/SEE
Continued on next page	

Figure 8. Generated course description in PDF.

3. Discussion

Before starting our market analysis, we looked at the state of the art and a first set of high level requirements that define our research criteria. These requirements are describing a subset of the program and course specification needs based on the University of Luxembourg.

Currently at the University of Luxembourg, Acrobat Reader, Microsoft Word, Excel and PowerPoint are widely used for describing, managing and assessing their programs. The program directors and teachers are often starting their description from scratch and without any method. Some universities may provide templates, that has to be filled out and sent to the administration office. This current approach has some advantages:

- familiar environment
- simple basic functionalities (filling out the cells, text formatting, etc.)
- simple installation
- export possibilities
- usable with many operating systems (Windows, MacOS, Linux, etc.)

and a lot of disadvantages:

- very costly (Acrobat Pro 15\$/user/month and Microsoft Office about 10\$/user/month)
- exhaustive specification task without templates
- lack of methods for specifying a program
- the use of Excel formula might be complicated even after following a training program
- collaborative and simultaneous work is impossible
- low performance if the document is very large and contain a large dataset
- entering large datasets manually to Excel is very time exhaustive
- inflexible auto format tools, bugs in format options
- it is very hard to maintain (no version control)
- loss of data in case of corrupted and damaged files

The University of Luxembourg provides MS Word templates for specifying the programs and registering the administration office. It is possible to describe a whole program and their courses using these templates. The instructors have to fill out the required fields and sent the specification to the secretary via email. Due to the missing specification methods, the teachers are free to write their own specification. They do not have any restriction on the content that may lead to some problems. Teachers often do not know how to describe their programs resp. courses. Most institutions do not even organize training sessions for their teaching staff. The following problems may occur:

- incomplete and unstructured program and course specification
- misunderstandings in between the university and the program/course holder
- misunderstandings in between learners and program/course holders
- incomparable documents
- not standardized to some international learning standard
- old and not maintained course specifications

For managing the course runtime and distributing the teaching material, the university of Luxembourg is using Moodle. Moodle is an open-source learning management system. Programs and courses may be described inside the tool. The teachers share information and course material using that platform.

Thanks to our analysis of the state of art, we were able to define a first set of high level requirements, that helped us to retrieve tools that could cover our needs. We were looking for tools that allow the static specification of institutions, programs and courses, dynamic specification during the course runtime, maintainable program specifications, the possibility to specify a program according to some international learning standard for accreditation purposes and the possibility to generate specification documents that may be distributed to the interested parties.

4. Materials and Methods

In this section, we aim at describing the materials and methods used to build the results, presented in Section 2, in order to allow potential reconstruction of our results by the readers (Note that the results from web search engines are unfortunately rarely reproducible, as their computation depends on a number of factors, as for instance the user performing the search, his/her geographical location, history of search, etc., most of these factors being totally unknown).

4.1. Surveys

4.1.1. Educational Programs Management Tools Survey

Here is the method that we have followed to construct the results for the educational programs management tools survey.

1. Firstly, the need for the survey has been identified and described in terms of the main objective presented in Section 2.1.1.
2. Then a survey protocol has been planned and executed for the primary study:
 - The strategy for searching is to use the Google search web engine with some keywords related to the TESMA domain: “tool” AND (“syllabus” OR “educational program” OR “education” OR “curriculum” OR “institution” OR “university” OR “school” OR “certification” OR “specification” OR “management”).
 - No specific criteria have been set to discard any data resulting from the search.
 - During this primary study the data extraction consisted in a textual listing of the website URLs of the resulting tools.
 - Two researchers of our team have performed this search. The result of this primary study has been the union of the two researchers’ results.
3. A secondary study has been planned and executed
 - The secondary study input were the results from the primary study, i.e., the list of websites URLs.
 - Some selection criteria have been defined, in particular, we excluded all tools from the primary study that
 - do not provide ways to specify educational programs, which is one of the main contribution of TESMA.
 - do not provide course runtime management, which is also an important feature of TESMA.
 - are e-learning platforms, which are not the focus of our project.
 - programs for which the coverage of TESMA feature categories were too low, i.e., less than 5% coverage.
 - The format of the extracted data has been performed systematically using a Google sheet “database” following this format, for each table’s row: the name of the tool, an operation provided by the tool, our feature category in which it can be mapped, a textual description of the operation, and the URL from which the description has been extracted.
 - The strategy used has been to
 - Study the online available documentation of the tool, e.g., user manuals or tutorials.
 - When available, we also installed the demo version of the tool and experimented it in order to extract implicit tool’s operations.
4. the data has been synthesized using pivot tables from the Google sheet table containing the extracted data. These pivot tables have then been inserted in the content of the present article in Tables 1, A1 and A3.

4.1.2. Educational Program Concepts Survey

Here is the method that we have followed to construct the results for the educational programs concepts survey.

1. Firstly, the need for the survey has been identified and described in terms of the main objective presented in Section 2.1.2.
2. Then a survey protocol has been planned and executed for the primary study:
 - We have identified ten universities with different cultures and locations.
 - We have explored their websites and documentations publicly available.
 - The data extraction has been performed systematically
 - We have analyzed the syllabi of each universities, focusing on software engineering educational programs, or more general computer science educational programs.
 - For each concept implicitly, or explicitly, used in the description of the program, we extracted the information and stored it in a Google sheet, that served us as a database.
 - each concept extracted, has been mapped, when possible to similar identified concepts of the TESMA requirements analysis model. Then also assigned to a category of concepts.
3. the data has been synthesized using pivot tables from the Google sheet table containing the extracted data. These pivot tables have then been inserted in the content of the present article in Tables 2 and A2.

4.2. TESMA Tool Set Development

We use the Messir methodology [2] for the specification of the TESMA tool set requirements. This methodology comprises a set of models (use-case, concept, environment, operation, etc.) for the specification of the requirements analysis software development phase. The specification is performed using a textual DSL and views may be created in a UML-fashion [6]. From the numerous types of models proposed in this approach, we mainly used in the context of this article, the use-case and the concept models types that allow on the one hand to specify the interaction of the system with the actors of its environment; and on the other hand to specify the domain concepts handled by our system (i.e., TESMA tool set).

5. Conclusions

In this article, we have presented in-depth surveys that showed us that we did not find a similar educational tool covering all our needs. The selected tools might not be applicable to different types of universities. This was a motivation for us to create our own tool that can be used by a large number of universities worldwide.

We have presented the current status of our tool development allowing educational institutions to specify educational programs and courses. The tool uses a textual domain-specific language with a graphical editor and includes a moodle generator that allows the specifier to generate complete moodle pages for the diffusion of course information and material.

As future work, we plan to iterate the process presented in this article to stabilize the requirements and the tool design and implementation. We also plan to study the automated generation of a web application from the language grammar in order to provide a user-friendly front-end that is mapped to our textual language grammar and provide a better coverage of our targeted TESMA features. We could take advantage of the translational semantics defined in the Messir methodology [2] to verify some properties of the TESMA models using the translated prolog specifications of our TESMA models. We plan to make our tool more generic and adaptable to the different kinds of terminologies used by universities over the world; this will involve the design and specification of a configurable grammar for our DSL. In order to ease the specification phase, we also plan to offer some libraries offering some kinds of templates that will be used as skeleton specifications directly in our tool.

Supplementary Materials: The Excalibur tool supporting the Messir [2] methodology is available online under the EPL v1.0 [14] open-source license at <http://messir.uni.lu>.

Acknowledgments: This study has been funded entirely by the University of Luxembourg. We would also like to thank the reviewers for their reviews that helped us improving the quality of this article.

Author Contributions: Nicolas Guelfi conceived and designed the experiments data structured; he also wrote a large part of the article's final content and participated in the processing of the reviewers' reviews. Benjamin Jahic worked on the data collection for the tools' market study, wrote the initial draft version of the article and participated in the writing of the reviewed version. Benoît Ries wrote the final version of the paper, collected data for the institutions concepts study, analyzed the data from the tools and institutions studies and participated in the writing of the reviewed version.

Conflicts of Interest: The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; nor in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

API	Application Program Interface
CS2013	Computer Science 2013 [15]
CSV	Comma-Separated Values
DSL	Domain-Specific Language
EMF	Eclipse Modeling Framework
EPL	Eclipse Public License
GNU	GNU is Not Unix
HTTP	HyperText Transfer Protocol
LLL	Long-Life Learning
PDF	Portable Document Format
PHP	PHP: Hypertext Preprocessor
REST	REpresentational State Transfer
SEE	Software Engineering Environments (course at University of Luxembourg)
SQL	Structured Query Language
SWEBOK	Software Engineering Body of Knowledge
TESMA	Tool for the Specification, Management and Assessment of educational programs
UML	Unified Modeling Language
WebApp	Web Application

Appendix A. Tool Market Analysis

Table A1. List of surveyed tools during the secondary study.

	Tool Name	Founder	License Type	Reference
1	TESMA	University of Luxembourg	open source	[3]
2	SchoolTool	Shuttleworth Foundation	open source	[16]
3	Moodle	Moodle	open source	[17]
4	Sakai	Apereo Foundation	open source	[18]
5	Open Education Resource	OEDb	proprietary	[19]
6	Ariadne Moodle Search Widget	Ariadne Foundation, Eummena	unknown	[20]
7	Responsive Open Learning Environments	Ariadne Foundation	open source	[21]
8	Learning Resource Exchange	European Schoolnet	proprietary	[22]
9	Atutor	Atutor	open source	[23]
10	Canvas	Canvas LMS	proprietary	[24]
11	Claroline	Claroline Connect	open source	[25]
12	Curriki	Curriki	proprietary	[26]
13	D2L	Desire2Learn	proprietary	[27]
14	EduCommons	EduCommons	open source	[28]
15	Joomla LMS	Joomla	proprietary	[29]
16	LAMS	LAMS Foundation	open source	[30]
17	LRN	LRN	open source	[31]
18	Olat	University of Zurich	open source	[32]
19	Open Tapestry	Open Tapestry	unknown	[33]
20	Fedena	Foradian	proprietary	[34]
21	OSM	University of Alabama	proprietary	[35]
22	Concourse-Syllabus Management	Intellidemia	proprietary	[36]
23	Alma	Alma	proprietary	[37]

Table A1. Cont.

	Tool Name	Founder	License Type	Reference
24	Schooltime	Schooltime	proprietary	[38]
25	Educube	Educube Division	proprietary	[39]
26	UOnline	University of NewCastle	unknown	[40]
27	Blackboard	Blackboard	proprietary	[5]

Appendix B. International Educational Concepts Analysis

Table A2. List of the selected institutions in the educational program concept survey.

	Institution Name	Educational Programs Surveyed	Website
1	University of Luxembourg	bachelor, master	[1]
2	University of McGill	bachelor, master	[41]
3	University of Southern California	bachelor, master	[42]
4	University of Geneva	bachelor, master, long-life-learning	[43]
5	University of Rennes 1	bachelor, master	[44]
6	The University of New South Wales	bachelor, master	[45]
7	Innopolis University	bachelor, master	[46]
8	King Abdulaziz University	bachelor, master	[47]
9	The Hong Kong University of Science and Technology	bachelor, master	[48]
10	National University of Singapore	bachelor	[49]

Appendix C. TESMA Feature Categories

Table A3. TESMA feature categories.

Category	Subcategory	SubSubcategory	T ₁	T ₂	T ₃	T ₄	T ₅	T ₆	T ₇
archive system	course history	course history			x	x			x
		student history						x	x
backend operation	backend operation	backend operation	x		x	x	x	x	
course	course	course				x			x
		participation management	x		x	x		x	x
	course description	course description			x	x			x
		dynamic course evaluation							x
		dynamic task management		x	x	x	x	x	x
		learning targets	x			x			x
		lecture description				x	x		x
		textual description	x			x			x
	course resources	book				x			x
		course creator						x	x
		digital media				x		x	
		external package				x			
		hardware		x	x	x	x		x
		human grouping		x	x	x	x	x	x
learner					x			x	
teacher							x	x	
file management	file explorer	file explorer				x			
		file handling operation	x				x		
		one level grouping	x		x			x	x
		reflexive grouping				x			x
generation	documentation	documentation			x			x	x
		evaluation report			x				x
		user report				x			x
		excel table generator							x
		moodle page generator							x
		participants statistics							x
	grading statistics				x	x	x	x	
statistics								x	
geographicallocation	geographical location	geographical location						x	
gui	assistance	assistance	x		x	x	x		
		highlighting			x	x			
	performance	performance			x		x	x	

Table A3. Cont.

Category	Subcategory	SubSubcategory	T ₁	T ₂	T ₃	T ₄	T ₅	T ₆	T ₇
institution	institution	institution							x
		institution description							x
interactions	intercommunication	contact management			x	x		x	
		intercommunication			x	x			
		mailing		x	x	x	x	x	
		real-time conversation						x	
	knowledge transmission	knowledge collection			x	x			
		knowledge exchange	x	x	x	x	x		
	notification	alert			x	x	x		x
		announcement	x		x	x	x		x
	submission	auto evaluation						x	x
		categorized grading	x	x		x			x
		customizable grading		x	x	x	x		x
		graded evaluation		x	x	x	x		x
		learning training		x		x	x		x
		numerical grading							
overall grading					x	x	x	x	x
ranked evaluation					x				x
learning standard	learning standard	learning standard desc.			x			x	
multi-platform	multi-platform	multi-platform				x		x	x
		textual editor							x
		webapp							x
		sirius							x
		office							x
other	other	other			x	x	x		
program	program	program				x			x
		program description							x
		program summary							x
		program organization							x
		courselist							x
time management	calendar	calendar			x	x			x
		course time management		x	x			x	x
		evaluation time management						x	x
		event time management			x	x	x		x
		learning time management				x			x
		task time management					x		x
	time management	time management							x
		semester planning							x
		lecture planning							x
									x
tool operations	automatised operation	calendar creation			x		x	x	x
		operation course handler	x		x	x	x		x
		operation evaluation handler		x	x	x			x
		operation grading handler	x	x	x	x	x	x	x
	importation	operation user handler	x	x	x	x		x	x
		external course description		x		x			x
		external hardware	x		x				
		external software		x	x	x	x		
	information retrieval	external user data	x		x				
		external course evaluation							x
		information retrieval			x	x	x		x
		find course							x
		find program							x
		find semester							x
tool requirements	language	language			x	x	x	x	x
	performance	performance				x			x
	scalability	scalability				x	x		x
user access rights	evaluation access rights	evaluation access rights				x			x
	file access rights	file access rights					x		
	user access rights	user access rights			x	x		x	x
	user authentication	user authentication				x	x		x
user management	user management	actor			x				x
		user information							x
		user details							x
		user location							x
		user management				x			x

Legend of Table A3: T₁: Canvas; T₂: D2L; T₃: Fedena; T₄: Moodle; T₅: Sakai Project; T₆: Schooltool; T₇: TESMA.

Appendix D. TESMA Conceptual Framework

Here is the first version of the conceptual framework related to TESMA. It comprises 85 concepts used to performed the institutions survey in Section 2.1.2:

Table A4. TESMA Concepts.

Category	Subcategory	Concepts
course	course description	course test rational task learningOutcome courseCredit courseID coursePeriod coursesModule courseSWEBOKCoverage courseType courseWorkload coueseRequirements fee informationComponent informationComponentOrdering iscdCode peopleNameComponent subject subjectID subjectRole syllabus
	course resources	artifact
file management	file explorer	courseSelection institutionCategory variableCategory variableNature variableType
generation	documentation	description reports
geographical location	geographical location	address country countryCode countryName gpsLocation latitude longitude
institution	institution description	institution

Table A4. Cont.

Category	Subcategory	Concepts
interactions	intercommunication knowledge transmission submission	email support categoricalValue nominalVariable variableCategoricalValue variableNumericalValue weight grade testID testType artifactScope taskType continuousVariable coursegrade discreteVariable groupgrade individualgrade measuredVariable modulegrade numericalVariable ordinalVariable overallgrade periodGraveAverage semestergrade
learning standard	learning standard	field fieldCoverage
program	program	educationalProgram educationalProgramType
time management	calendar time management	date testDate milestone period time
tool operations	Create-Read-Update Delete operations	user operations
tool requirements	language	language
user management	user management user description	actor board group promotion couseStudents student instructor addressComponent firstName lastName officeHours studentID

References

1. University of Luxembourg Website. Available online: <http://wwwen.uni.lu> (accessed on 30 November 2016).
2. Guelfi, N. The Messir Scientific approach to Requirements Engineering. In *Laboratory for Advanced Software Systems Technical Report TR-LASSY-16-01*; University of Luxembourg: Luxembourg, 2016.

3. Guelfi, N.; Jahić, B.; Ries, B. TESMA: Towards the Development of a Tool for Specification, Management and Assessment of Teaching Programs. In Proceedings of the 2nd International Conference on Applications in Information Technology, Balikpapan, Indonesia, 26–27 October 2016; pp. 5–8.
4. Kitchenham, B.; Charters, S. *Guidelines for performing Systematic Literature Reviews in Software Engineering*; Ebse Technical Report; Keele University and Lincoln University: Lincoln University, PA, USA, 2007.
5. Blackboard. Available online: <http://uki.blackboard.com/> (accessed on 30 November 2016).
6. ISO/IEC. *Information Technology—Object Management Group Unified Modeling Language OMG UML—Part 2: Superstructure*; ISO/IEC 19505-2-2012; International Organization for Standardization: Geneva, Switzerland, 2012.
7. Kosar, T.; Bohra, S.; Mernik, M. Domain-Specific Languages: A Systematic Mapping Study. *Inf. Softw. Technol.* **2016**, *71*, 77–91.
8. Bettini, L. *Implementing Domain-Specific Languages with Xtext and Xtend*, 1st ed.; Packt Publishing, Incorporated: Birmingham, UK, 2013.
9. Gamma, E.; Beck, K. *Contributing to Eclipse: Principles, Patterns, and Plug-Ins*, 1st ed.; Addison-Wesley Professional, Incorporated: Boston, MA, USA, 2003.
10. Steinberg, D.; Budinsky, F.; Paternostro, M.; Merks, E. *EMF: Eclipse Modeling Framework*, 2nd ed.; Addison-Wesley Professional, Incorporated: Boston, MA, USA, 2008.
11. Jahić, B. Requirements, Design and Implementation of a Teaching Specification, Management and Assessment tool. *Laboratory for Advanced Software Systems Technical Report TR-LASSY-16-03*; University of Luxembourg: Luxembourg, 2016.
12. Tahchiev, P.; Leme, F.; Massol, V. *JUnit in Action*, 2nd ed.; Manning Publication: Greenwich, CT, USA, 2010.
13. Viyović, V.; Maksimović, M.; Perisić, B. Sirius: A rapid development of DSM graphical editor. In Proceedings of the IEEE 18th International Conference on Intelligent Engineering Systems (INES), Tihany, Hungary, 3–5 July 2014; pp. 233–238.
14. Foundation, E. Eclipse Public License—v 1.0. <https://www.eclipse.org/legal/epl-v10.html> (accessed on 30 November 2016).
15. Draft, S. *Computer Science Curricula 2013*, 1st ed.; ACM and IEEE Computer Society, Incorporated: New York, NY, USA, 2013.
16. SchoolTool the Global Student Information System Website. Available online: <http://schooltool.org> (accessed on 30 November 2016).
17. Moodle Website. Available online: <https://www.moodle.com> (accessed on 30 November 2016).
18. Foundation, A. Sakai Website. Available online: <https://sakaiproject.org> (accessed on 30 November 2016).
19. OEDb. Open Education Resource (OER) Website. Available online: <http://oedb.org> (accessed on 30 November 2016).
20. Eummena, A.F. Ariadne Moodle Search Widget Website. Available online: <http://www.eummena.org/index.php/labs> (accessed on 30 November 2016).
21. Foundation, A. Responsive Open Learning Environments (ROLE) Website. Available online: <http://role-project.archiv.zsi.at/index.html> (accessed on 30 November 2016).
22. Schoolnet, E. Learning Resource Exchange (LRE) Website. Available online: <http://lreforschools.eun.org/web/guest/about> (accessed on 30 November 2016).
23. Atutor Website. Available online: <http://www.atutor.ca/atutor/features.php> (accessed on 30 November 2016).
24. Canvas Website. Available online: <https://www.canvaslms.com> (accessed on 30 November 2016).
25. Connect, C. Claroline Website. Available online: <http://www.claroline.net> (accessed on 30 November 2016).
26. Curriki Website. Available online: <http://www.curriki.org/> (accessed on 30 November 2016).
27. Desire2Learn. Available online: <https://www.d2l.com/> (accessed on 30 November 2016).
28. EduCommons. Available online: <https://sourceforge.net/projects/educommons/> (accessed on 30 November 2016).
29. Joomla LMS Website. Available online: <https://www.joomlals.com> (accessed on 30 November 2016).
30. Foundation, L. LAMS Website. Available online: <https://www.lamsfoundation.org> (accessed on 30 November 2016).
31. LRN Tool Website. Available online: <http://www.dotlrn.org> (accessed on 30 November 2016).

32. Olat Website. Available online: <http://www.olat.org> (accessed on 30 November 2016).
33. Tapestry, O. Open Tapestry Website. Available online: <http://www.opentapestry.com> (accessed on 30 November 2016).
34. Fedena Website. Available online: <http://www.fedena.com> (accessed on 30 November 2016).
35. OSM Website. Available online: <http://syllabi.ua.edu/> (accessed on 30 November 2016).
36. Intellidemia. Concourse–Syllabus Management Website. Available online: <http://www.intellidemia.com> (accessed on 30 November 2016).
37. Alma Website. Available online: <http://www.getalma.com> (accessed on 30 November 2016).
38. Schoolltime. Available online: <https://school-time.co/> (accessed on 30 November 2016).
39. Division, E. Educube Website. Available online: <http://www.educube.net> (accessed on 30 November 2016).
40. UOnline Website. Available online: https://uonline.newcastle.edu.au/webapps/portal/execute/tabs/tabAction?tabId=_26455_1&tab_tab_group_id=_181_1 (accessed on 30 November 2016).
41. University of McGill Website. Available online: <http://www.cs.mcgill.ca> (accessed on 30 November 2016).
42. University of Southern California Website. Available online: <http://www.usc.edu> (accessed on 30 November 2016).
43. University of Geneva Website. Available online: <http://www.unige.ch> (accessed on 30 November 2016).
44. University of Rennes 1 Website. Available online: <https://www.univ-rennes1.fr> (accessed on 30 November 2016).
45. The University of New South Wales Website. Available online: <https://www.unsw.edu.au> (accessed on 30 November 2016).
46. Innopolis University Website. Available online: <https://university.innopolis.ru/en/> (accessed on 30 November 2016).
47. King Abdulaziz University Website. Available online: http://www.kau.edu.sa/home_ENGLISH.aspx (accessed on 30 November 2016).
48. The Hong Kong University of Science and Technology Website. Available online: <http://prog-crs.ust.hk> (accessed on 30 November 2016).
49. National University of Singapore Website. Available online: <http://www.comp.nus.edu.sg> (accessed on 30 November 2016).



© 2017 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).