Applied Computer Systems
ISSN 2255-8691 (online)
ISSN 2255-8683 (print)
December 2020, vol. 25, no. 2, pp. 134–144
https://doi.org/10.2478/acss-2020-0015
https://content.sciendo.com

Definition and Validation of the Subset of SCORM Requirements for the Enhanced Reusability of Learning Content in Learning Management Systems

Sintija Petrovica[1*], Alla Anohina-Naumeca[2], Andris Kikans[3]
[1, 2] *Riga Technical University, Riga, Latvia*
[3] *Datorzinibu centrs Ltd., Riga, Latvia*

Abstract **– Nowadays, interoperability of learning management systems is still not very high. The authoring tools can help transfer e-learning content between different learning management systems. However, in this context, they should be able to produce learning content that is compliant with some industry standards. One of the most widely used standards is the SCORM 1.2 release. The research addresses the extension of the functionality of the previously developed content development tool EMMA by incorporating into it the support for the subset of SCORM 1.2 requirements. The paper describes the process of the acquisition, implementation, and validation of the defined requirements. Moreover, it presents the results of the analysis of 33 SCORM authoring tools and 16 SCORM players.**

Keywords **– E-learning, interoperability, learning management system, Moodle, SCORM 1.2, SCORM authoring tool, SCORM player.**

I. Introduction

Automation of the acquisition of knowledge and skills was one of the first areas for the use of computerised solutions. Nowadays, when information and communication technologies have thoroughly intruded into everyday human life, the development and the wide use of computerised learning tools and platforms are a natural and unavoidable consequence. In this context, Global Industry Analysts, Inc. [1] forecasts the global e-learning market to reach 305.3 billion U.S. dollars by 2025. Other sources of information mention that it could surpass 243 billion U.S. dollars by 2022 [2] or, between 2018 and 2023, its compound annual growth will rise by 10.26 %, reaching a total market size of around 286.62 billion U.S. dollars compared to 159.52 billion U.S. dollars in 2017 [3]. In Europe, this market is also expected to grow by more than 24.23 billion U.S. dollars between 2020 and 2024 [4]. The adoption of e-learning in both the academic and corporate sectors is based on many advantages offered by this type of learning/training, including opportunities to meet different learning needs, cost-effectiveness, access to learning content on an unlimited basis, promotion of active and independent learning habits, and many others [1].

Gilbert [5] points out that today there are around 700 vendors of learning management systems (LMSs), and this market could be one of the largest in the entire technology industry in the immediate future. At the same time, there is one of the most significant challenges in this market, which is considered to be the most desired improvement in future LMSs – an ability to integrate with other systems [3]. In this context, ensuring the interoperability of LMSs becomes the absolute necessity.

Unfortunately, the potential economic benefits of the development of LMSs lead to a situation when the developers of such systems competing for the market offer different e-learning content delivery environments [6]. This reduces the interoperability of the systems and, as a result, makes difficult the transfer of high-cost e-learning content from one LMS to another or distributing the content widely [6], [7]. Thus, when updating or changing the LMS, the content developer should abandon the expensive content that has already been created and start creating it from scratch or should develop completely different versions of the content corresponding to different LMSs [6]. This demands to make the learning content reusable, ensuring the interoperability of LMSs and introducing standardisation in this area. One of the most common e-learning standards is SCORM [8]. Since its first version released in 2000, the SCORM standard has continuously evolved. However, according to [9], the SCORM 1.2 release is still the most widely supported in a variety of e-learning products and solutions. Therefore, the ability of content development tools, the so-called authoring tools, to produce the SCORM-compliant learning content is a vital competitive advantage.

The current research focuses on the extension of the functionality of the content development tool EMMA by incorporating into it the support for the subset of SCORM 1.2 requirements. The mentioned tool has been developed and owned by a Latvian enterprise "Datorzinibu centrs" since 2013. The abbreviation EMMA means the electronic learning material aggregation tool in the Latvian language (*Elektronisko Mācību Materiālu Agregators*). The EMMA's architecture complies with the logical model of e-learning materials [10], which includes three object tiers – e-learning materials, e-

[*] Corresponding author's e-mail: sintija.petrovica@rtu.lv

learning objects, and elementary e-learning objects, which can be combined in different structures, support interrelations and can be described by rich metadata. The EMMA tool allows creating multimedia-rich and interactive learning content in HTML5, which can be further used to generate a source code of learning content for specific technology solutions – LMSs, mobile applications, printable format, and/or webpages. In addition, the tool provides high-level customisation flexibility in terms of adding new content designs and layouts using standardised programming tools. Moreover, the EMMA tool has the following advantages – availability of a source code and documentation, freeware, both cloud-based and on-premises deployment options, and the simple content definition process with both human-readable file import and visual mode.

The paper presents the process and results of the definition and validation of the subset of SCORM 1.2 requirements to enhance the functionality of the EMMA tool and to achieve higher reusability of learning content in LMSs. Moodle LMS is the central platform targeted by this research project. The structure of the paper is as follows. The next section provides a short overview of the SCORM standard. After that, the process of defining the subset of SCORM 1.2 requirements is described together with the requirements acquired. Later, the extension of the functionality of the EMMA tool and its validation are presented. Conclusions are given at the end of the paper.

II. SCORM Standard

SCORM or Sharable Content Object Reference Model can be better described when focusing on separate parts of its title:

- "Sharable Content Object" corresponds to online learning content shared across different systems [11], [12].
- "Reference Model" points out that it is not a standard in its direct understanding, but rather the integration of results achieved by other standardisation organisations with reference to existing standards [11], [13], [14].

The SCORM standard defines the technical aspects related to the structure of learning content [15] and also the way the LMS delivers this content [11], [15]. Therefore, it does not deal with the learning process and pedagogical principles of the content organisation [15].

The release 1.2 of the SCORM standard includes the following constituent parts [13], [16]:

1) Content Aggregation Model (CAM) "promotes consistent storage, labelling, packaging, exchanging and discovery" of e-learning content [12];
2) Run-time environment (RTE) is a JavaScript-based mechanism, which provides communication between learning content and LMS through the exchange of data elements [16], [17].

One of the central concepts of the SCORM standard is a SCORM package. A SCORM package is a collection of all files that are needed for launching learning content [14]. They are specified by the CAM that incorporates several components: Content Model, Meta-data, and Content Packaging. The Content Model describes components of learning experience and their organisation [12]. The components include the smallest units called assets (digital representations of images, sound, HTML pages, text data, etc.) that can be further combined into sharable content objects (SCOs) that can be aggregated and then used to build organisations [18]. The assets do not communicate with an LMS [19]; however, they are delivered to students [20]. SCOs are able to communicate with an LMS and they can be launched by the student who uses an LMS [21]. The organisations provide a mapping between SCOs and learning activities [19].

Meta-data describes properties of the SCORM package, including the history and current state of the resource, technical requirements, educational characteristics, the intellectual property rights and conditions of use, the relationship between the resource and other resources, etc. [12], [16]. Content Packaging allows implementing an exchange of learning content between different systems [12]. The SCORM standard requires all files of learning content to be transferred in a folder or a .zip file corresponding to Package Interchange File format [16]. The files contain both a manifest file and physical files that correspond to learning resources [14]. The manifest file describes metadata of a SCORM package, structure of learning resource, and resource-supporting files included in the package.

The RTE consists of three components [12], [22]:

1) Launch mechanism, which defines a standard way of launching SCOs by LMSs;
2) Application Programming Interface (API) that supports communication between an LMS and SCOs facilitating interoperability and reusability of learning content;
3) Data model, which contains a standardised set of elements used to describe SCOs.

According to [16], the API is implemented by an API adapter that is an ECMAScript (JavaScript) object with the title "API" and it does not support other communication ways with LMS like web servers or HTTP requests. The adapter should implement eight functions [16]:

- *LMSInitialize()* – it opens the connection between the SCO and LMS;
- *LMSFinish()* – it closes the connection between the SCO and LMS;
- *LMSGetValue()* – it retrieves a value of an element of the data model from the LMS;
- *LMSSetValue()* – it writes a value for an element of the data model to the LMS;
- *LMSCommit()* – it ensures that the data are persisted after any values are set;
- *LMSGetLastError()*, *LMSGetErrorString()* and *LMSGetDiagnostic()* – they handle errors in communication with LMS.

Data model elements can be read and written using the API [13]. They specify what can be stored in and retrieved from an LMS. The elements are divided into two categories: mandatory and optional. To recognise an LMS as a SCORM-compliant system, it should provide support to all mandatory elements.

III. Defining the Subset of SCORM Requirements

This section presents the process of defining the subset of SCORM requirements to extend the functionality of the EMMA. It specifies the methodology used and the results achieved.

A. Methodology

The acquisition of the requirements involved three steps.

1) The analysis of an overview included in the Technical Specification of SCORM 1.2 standard [13] and Conformance Requirements of the mentioned standard [23] to identify requirements for SCORM 1.2 compliant learning content (both mandatory for running SCOs and supporting additional features, e.g., knowledge assessment and multiple learning paths).
2) The examination of the existing SCORM 1.2 authoring tools and SCORM players to determine general and functional features of such tools and players as well as to identify the most common features that should be supported by similar tools.
3) The analysis of the specification of the EMMA tool to understand its current functionality and to define requirements for its extension, considering results achieved in two previous steps.

The authoring tools mentioned in Step 2 were selected based on the available lists of SCORM authoring tools on web pages describing the Moodle environment, i.e., [24], [25]. Besides, the list of tools offered in the SCORM standard description [26] and tools mentioned in scientific publications (e.g., [27]) were explored. As a result, 33 tools were identified, and 23 functional features were extracted based on their analysis.

The selection of SCORM players in Step 2 considered the following aspects:

- the player supports SCORM 1.2 standard;
- the developers still support the player (a link to the player is working);
- the link to player's installation files or source code is available;
- the player is fully operational (it is not under development);
- the player is a general-purpose solution (it is not created for a particular application or an LMS).

As a result, 16 SCORM players were found and analysed allowing us to identify 31 functional and general features supported by them.

B. Analysis of SCORM 1.2 Standard

During the analysis of the SCORM standard, mandatory and optional conformance requirements for SCOs and content packaging were studied [23], as well as requirements for the description, organisation, and delivery of learning content were explored [13]. As a result, four requirements were defined allowing one to ensure:

- launch of SCOs (for the first time or repeatedly after it has been paused or closed);
- implementation of interactions for knowledge assessment;
- assessment of interactions;
- navigation and sequencing of learning content.

During the definition of requirements, particular attention was paid to the elements of the data model, which are part of the SCORM RTE [13] and are intended to support the implementation of a specific requirement.

Requirements for the launch of SCOs – the launch process of SCO can be divided into several steps. These steps include the determination of SCO to be launched, setting its running mode, initiating SCO data and place to start it, SCO interaction with the learner and tracking his/her learning progress, saving all the data related to the SCO mastery.

Requirements for the implementation of interactions – SCORM 1.2 standard uses a category of elements of the data model starting with *cmi.interactions*. Interactions represent different types of learner's responses that can be recorded during the interaction with the SCO. The elements in this category allow collecting the detailed information about each input while the learner acquires the SCO. In the SCORM 1.2 standard, eight categories of interactions are available [13]: "*true-false*", "*choice*", "*fill-in*", "*matching*", "*performance*", "*sequencing*", "*likert*" and "*numeric*", which are defined in the element of the data model *cmi.interactions.n.type* by the SCO. It should be noted that SCORM 1.2 does not forbid the use of other types of interactions.

Requirements for the assessment of interactions – several approaches can be used to assess interactions or knowledge assessment tasks (e.g., tests). The first approach uses elements of the data model already defined for interactions. These elements store both the correct answers (the so-called patterns *cmi.interactions.n.correct_responses.n.pattern*) and the answers provided by the learner allowing for the comparison of answers. The second approach is based on the results of the first approach, as it allows assessing whether the result of answer comparison meets the minimum required for the specific interaction to be not only completed but also passed. In this case, the mastery score defined in the manifest file (an element <*adlcp:masteryscore*>) is used as a standard against which the result is compared.

Requirements for the navigation and sequencing of learning content – the sequence of learning resources and the navigation between them allow expanding possibilities of learning resources by adding mechanisms that take into account the conditions of navigation from one learning resource to the next [13]. SCORM 1.2 standard does not provide special control mechanisms for the transition between learning resources; however, two simple approaches can be applied. The first approach includes a definition of multiple organisations for learning content to create several sequences for learning resources [13]. The second approach uses prerequisites defined in the manifest file (the element <*adlcp:prerequisites*>) for learning resources indicating on other resources in the organisation that must be completed before a particular learning resource [13].

All the requirements listed above are either functionally necessary for running SCOs (e.g., the launch of SCOs) or support the assessment of learner's knowledge and provision of several learning paths. This leads to the development of interactive learning content with a possibility to personalise the learning process (e.g., through adapted learning paths and assessments) and follow the learning progress. Therefore, the implementation of these requirements is important within the frame of the current study and should be considered during the extension of functionality of the EMMA tool and features provided.

C. Examination of SCORM 1.2 Authoring Tools and Players

The creation of content packages can be a rather tricky task for learning content developers, especially when it comes to defining the manifest file and metadata. Therefore, the authoring tools can be used to create SCORM-compliant learning content, which in most cases can be exported as an archive in .zip format. These tools can be divided into two general categories: 1) tools that allow creating SCORM-compliant learning content packages but do not support the development of learning resources and 2) tools that allow for the development of SCORM-compliant learning content by transforming it into SCORM standard-compliant learning objects which can be further exported for the use in a variety of contexts and systems.

Table I presents the detailed summary of the tools examined in the study and their supported features. A list with the tools and their links is available in Appendix I.

TABLE I
FEATURES MOST FREQUENTLY IMPLEMENTED IN THE SCORM AUTHORING TOOLS EXAMINED IN THE STUDY

Features Tools	Free or free trial software tool	Content package export in .zip format	Support for multiple e-learning standards (SCORM 1.2, 2004, xAPI)	Import of existing learning resources	Use of a package independently or its integration into LMS	Data tracking (e.g., learners’results)	Multi-device support (responsive design)	Development of interactive webpages	Development of learning content using pre-designed templates or forms	Automatic generation of a manifest	Publishing learning content in HTML5 format	Content preview (WYSIWYG - What you see is what you get)	Development of interactive quizzes	Support for Flash format	Integration with MS Office tools	Support for multiple metadata models (e.g., Dublin Core, LOM, LOM-ES)	Development of complex interactions (e.g., simulations)	Online tool	Drag and Drop possibilities	Support for multiple operating systems	Communication testing with a learning management system	Editing existing content packages	Development of game-based content	Number of features supported by a tool
Adobe Captivate	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓		✓	✓			✓	✓			17
iSpring Free	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓								15
GOMO	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓						✓	✓		✓			14
Easygenerator	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	✓					✓	✓					13
ActivePresenter	✓	✓	✓	✓		✓	✓		✓	✓	✓	✓	✓				✓			✓				13
isEazy	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓						✓					✓	13
CourseLab	✓	✓	✓	✓	✓		✓	✓	✓	✓		✓				✓	✓		✓					13
Adobe Presenter	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓		✓	✓	✓									13
iSpring QuizMaker	✓	✓	✓	✓	✓	✓	✓		✓		✓	✓	✓		✓				✓					13
eXe Learning	✓	✓	✓	✓	✓		✓	✓		✓	✓	✓				✓				✓				12
Exerte	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓							✓						12
Lectora Online	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓							✓	✓						12
Articulate 360	✓	✓	✓	✓	✓	✓	✓				✓	✓	✓	✓	✓									12
ToolBook		✓	✓	✓	✓	✓	✓		✓			✓	✓		✓		✓							11
Wondershare Quiz Creator	✓	✓	✓	✓	✓	✓		✓	✓				✓	✓										10
eLearning Authoring Tool	✓	✓	✓	✓	✓			✓		✓				✓		✓			✓					10
Udutu Authoring Tool	✓	✓	✓	✓	✓		✓		✓	✓	✓	✓						✓						11
Simple SCORM Packager	✓	✓	✓	✓	✓	✓		✓		✓						✓						✓		10
ReadyGo Web Course Builder	✓	✓	✓	✓		✓	✓		✓	✓	✓		✓											10
MS LCDS	✓			✓	✓	✓		✓	✓			✓			✓		✓					✓		10
Reload Editor	✓	✓	✓	✓						✓						✓			✓	✓		✓		9
docendo	✓	✓	✓	✓	✓			✓				✓				✓		✓						9
MOS Chorus LCMS	✓	✓	✓	✓			✓	✓	✓	✓	✓													9
ViewletQuiz	✓	✓	✓	✓	✓					✓			✓	✓										8
Wondershare PPT2Flash	✓	✓	✓	✓	✓									✓	✓		✓							8
Coral Presenter	✓	✓		✓	✓	✓							✓	✓	✓									8
Helius Presenter	✓		✓	✓	✓								✓	✓	✓									7
Advanced SCORM Editor	✓	✓			✓					✓						✓				✓				6
ContentGenerator.net	✓	✓	✓			✓								✓									✓	6
SCORM Developer Tools	✓		✓		✓			✓						✓							✓			6
PowerQuizPoint Quiz Creator	✓				✓						✓		✓	✓	✓									6
THESIS Rapid SCORM eLearning	✓	✓	✓												✓									4
Scormify	✓	✓		✓			✓																	4
Number of tools supporting a particular feature	32	29	28	28	26	18	18	17	17	15	14	14	13	12	11	9	7	7	6	5	3	3	2	

To define the subset of SCORM requirements within the framework of the current research, nine features supported by more than half of the authoring tools examined in the study are considered:

1) free or free trial software tool;
2) content package export in .zip format;
3) support for multiple e-learning standards (SCORM 1.2, SCORM 2004, xAPI);
4) import of existing learning resources;
5) use of a content package independently or its integration into LMS;
6) data tracking (e.g., learners' results);
7) multi-device support (the use of responsive design);
8) development of interactive webpages;
9) development of learning content using pre-designed templates or forms.

The main goal of SCORM players is to deliver learning content included in the content packages to learners. Table II displays the detailed summary of the SCORM players examined in the study and their supported features. In the table, those features that are supported by the players are marked with "✓", and features which are not supported – with "—". Empty cells mean that a particular feature was not mentioned or described in the documentation of a player. However, it cannot be asserted that the player supports or does not support a particular feature. A list with the players and their links is available in Appendix II.

TABLE II

FEATURES MOST FREQUENTLY IMPLEMENTED IN THE SCORM PLAYERS EXAMINED IN THE STUDY

Features Players	Free or free trial SCORM player	Tracking learners' progress	Integration of a player into a webpage	Support for all mandatory elements of the Data model	Integration of a player into LMS	Support for all 8 API functions	Stand-alone player	Available documentation	Playing of SCORM packages in offline mode	Reporting mastery score	Playing content within a frame	Support for multiple operating systems	Tracking API function calls and changes in the data model	Tracking interactions	Use of LocalStorage or Cookies for data storage	Support for multiple e-learning standards	Available source code	Available pause and resume options	Support for multi-SCO content packages	Multi-device support	Support for all optional elements of the data model	Playing content in a new window	Use of JavaScript	Playing content in full or adapted screen size	Tracking objectives	Possibility to integrate the player into Open edX	Support for prerequisites	Import of other file types (not only .zip)	Preview and exploration of package structure	Support for multiple organisations	Use of Java	Number of features supported by the player
Simple SCORM LMS Adapter (SSLA)	—	✓	✓	✓	✓	✓	—	✓	—	✓			✓	✓	✓	✓	✓	✓	✓		✓	✓	✓		✓	✓		—			—	19
Appsembler SCORM Xblock	✓	✓		✓	✓	✓	—	✓	—	✓	✓	✓	—	✓	✓	—	✓	✓	✓	✓	—	✓	—	✓		✓		—			—	18
SCORM xblock	✓	✓		✓	✓	✓	—	✓	—	✓	✓	✓	—	✓	✓	—	✓	✓	✓	✓	—	✓	—	✓		✓		—			—	18
Moodle app SCORM player	✓	✓		✓	✓	✓			✓	✓	✓	✓	—	✓		—		✓		✓	✓	—			✓	—	✓	—				15
SimpleAPI	✓		✓	✓		✓	✓	✓	✓	✓		✓	—	✓	✓	—			—		✓		✓		✓	—		✓				15
SCORM 1.2 Test Wrap	✓	✓	✓		✓		✓	✓			✓		✓	✓	✓	—		✓				✓	✓		✓	—		✓				15
VSSCORM (Very Simple SCORM Interface)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	—		—	✓		✓		—	—	✓		—	—					—	15
TinyLMS	✓		✓	✓	—		✓	✓	✓		✓	✓			✓	—	✓		✓		—	—	✓			—		—	✓	✓	✓	15
RELOAD SCORM 1.2 Player	✓	✓	✓	✓			✓	✓	✓		✓	—	✓			—			✓	—	✓	—				—	✓			✓		13
Raccoon Gang SCORM Integrator xBlock	✓	✓	✓	✓	✓	✓		✓	—	✓			—	✓		✓	✓				—		—	✓		✓		—			—	13
SCORM Proxy Player (SCORMPool)	✓		✓	✓			✓	—	✓	✓			✓		✓	✓	—	✓			✓					—	✓					12
Rustici Engine	—	✓	✓		✓	✓	✓		✓			✓				✓	—			✓						—		✓			✓	11
Scorm Preview Ninja	✓		✓		—		✓	—	—		✓	✓	✓			✓	—			✓		—		✓		—		—			—	9
Offline SCORM Player	—	✓	✓		✓		✓		✓			✓				✓	—			✓						—		—			—	8
SCORM Cloud	✓	✓	—		✓	✓	—		—				✓			✓	—					✓				—						7
SCORM Player	✓											—												✓		—		—	✓			3
Number of players supporting the feature	13	11	11	10	10	9	9	9	8	8	8	8	7	7	7	7	6	6	6	6	5	5	5	5	4	4	3	3	2	2	2	

To define the subset of SCORM requirements within the framework of the current research, eight features supported by more than half of the SCORM players are further evaluated:

1) the free or free trial SCORM player;
2) tracking learners' progress;
3) integration of the player into a webpage;
4) support for all mandatory elements of the data model;
5) integration of the player into LMS;
6) support for all 8 API functions;
7) the stand-alone player (no need to integrate it into a webpage);
8) available documentation.

After the thorough analysis of the features previously identified (acquired both from the examination of SCORM authoring tools and players), the key features influencing the development of learning content and interaction with the content were selected by the research team for further consideration in terms of the EMMA tool. The selected features are:

- directly related to the development of learning content (e.g., import of learning resources, creation of interactive, responsive and SCORM-compliant learning content, and its export in various formats);
- related to the use of the learning content (independently or within LMS or webpage);
- related to data tracking (i.e., learner's progress while mastering SCOs).

D. Analysis of EMMA Software Specification

The features acquired in the previous steps (from the analysis of SCORM 1.2 standard, authoring tools and SCORM players) were considered during the examination of the software specification of the EMMA tool. The goal was to review the tool functionality and propose its extension with the aim of supporting the SCORM 1.2 standard.

The process of content development in the EMMA tool is based on two mutually interacting models. The logical learning content model [10] includes elementary units (like text, audio, video, etc.) which may be combined into larger content objects. These source data are ready for transformation into HTML5 to be further used in the generation of content source code, i.e., xAPI. The conditional model or Razor is represented as CSHTML files and applies to the whole logical model or its subsets. This model is used to convert the content source code into operational learning content by supplementing it with CSS and JavaScript. The model conversion process is shown in Fig. 1.

Currently, the EMMA tool does not support the SCORM 1.2 standard. Based on the analysis of the tool software specification and its current functionality, various features were identified, e.g., creation of learning content of different degrees of complexity, adding and editing of metadata necessary for the discovery of learning content and use of HTML5 for the creation of multimedia-rich learning content. These features were combined with the requirements previously defined from the examination of SCORM 1.2 standard and features supported by SCORM authoring tools and content players to extend functionality of EMMA. Apart from the requirements functionally needed to launch the SCORM-compliant learning content, overall nine requirements for the extension of EMMA functionality were identified (most of them are already partly supported by the EMMA tool) to ensure full compliance with SCORM 1.2 standard. The distinguished requirements allow providing:

- support for several content organisations;
- different degrees of aggregation of learning content;
- import of earlier created learning resources to the content;
- a visual preview of HTML code;
- support for knowledge assessment activities;
- metadata adding and editing;
- export of learning content as an archive;
- creation of HTML5 based learning content;
- export of a content package for the independent use (without LMS).

Requirements for the support for several content organisations – currently, the tool already provides the support for different content organisations, including displaying the content hierarchy, retrieving the content structure in JSON format, and editing the structure. This is also an option supported by SCORM 1.2 standard allowing for the creation of multiple sequences of the same learning resources and defining the navigation between resources.

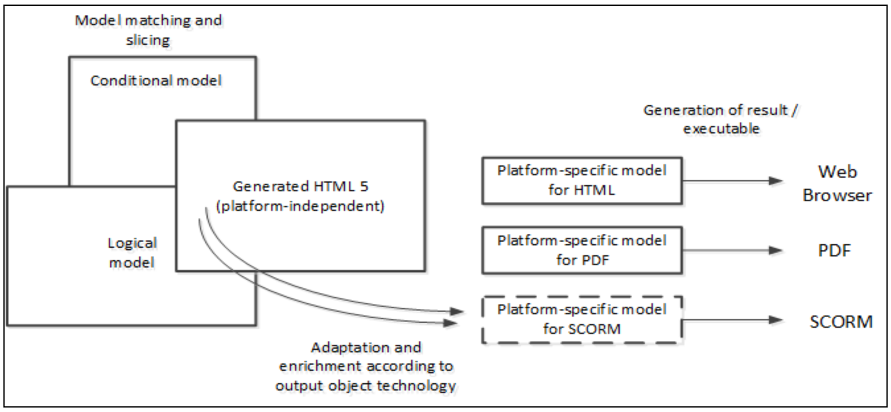

Model matching and slicing
Conditional model
Generated HTML 5 (platform-independent)
Logical model
Generation of result / executable
Platform-specific model for HTML
Web Browser
Platform-specific model for PDF
PDF
Platform-specific model for SCORM
SCORM
Adaptation and enrichment according to output object technology

Fig. 1. The model matching and conversion process in the EMMA tool.

Requirements for different degrees of content aggregation – the requirement was defined based on the analysis of EMMA specification. The tool already provides three levels of learning content aggregation. In this case, there is a great similarity with the components of the SCORM-compliant components, which can be arranged from the smallest component to the largest to create the SCORM-compliant learning content.

Requirements for the import of earlier created learning resources to the content – the tool provides a possibility to add the learning resources previously created to the developed learning content. This was one of the most common features identified during the analysis of other authoring tools.

Requirements for a visual preview of HTML code – the tool already provides a visual preview of HTML code. This possibility should also be ensured by developing SCORM-compliant learning content. It makes the creation of learning content easier because of the possibility to see how the created content will look like. Such a feature was also distinguished in other authoring tools.

Requirements for the support for knowledge assessment activities – currently, the tool allows for the creation of tests with one or multiple correct answers. At the same time, the SCORM standard provides eight different categories of interactions.

Requirements for metadata adding and editing – the metadata play an essential role in the SCORM standard, so it is important to add and edit them according to the IEEE LOM standard. Based on the analysis of EMMA specification, it can be concluded that the tool implements an almost complete set of mandatory metadata; however, it lacks the category "*Relations*" that describes resource relationships with other resources. This requirement therefore is directly related to the development of learning content and use of learning resources.

Requirements for the export of learning content as an archive – the tool needs to provide an export of learning content in .zip archive, including learning content in HTML format, files and metadata in XML format to ensure that learning content can also be used in LMSs, which support SCORM packages only in the archive format.

Requirements for the creation of HTML5 based learning content – the tool already supports the development of HTML5 based learning content. This was an important aspect of the analysis of other authoring tools enabling users to take an advantage of the latest multimedia files and technologies and improving learning content interactivity and accessibility both in computers and mobile devices.

Requirements for exporting and using learning content independently of LMS – it is necessary to ensure that content packages can be exported for the further use independently of other systems (e.g., LMSs) allowing the user to run learning content easily on a web page. The SCO itself should be able to provide an assessment of its acquisition in such a case. The learner's progress and achievements should be stored in a file that the object can access (read and write).

IV. Extension of the Functionality of the EMMA Tool

Based on the subset of SCORM requirements defined in the previous section, initial improvements in the EMMA tool have been already made to ensure that the learning content produced by the tool is fully compatible with the most popular LMSs (e.g., Moodle, Open edX, etc.). There were three main improvement areas where the further extension of the tool functionality was mandatory:

1) requirements for the support for knowledge assessment activities (i.e., interactions);
2) requirements for the export of learning content as an archive;
3) requirements for exporting and using learning content independently of LMS.

In regards to the implementation of additional knowledge assessment activities, it was necessary to introduce seven categories of *cmi.interactions* – "*true-false*", "*fill-in*", "*matching*", "*performance*", "*sequencing*", "*likert*", and "*numeric*".

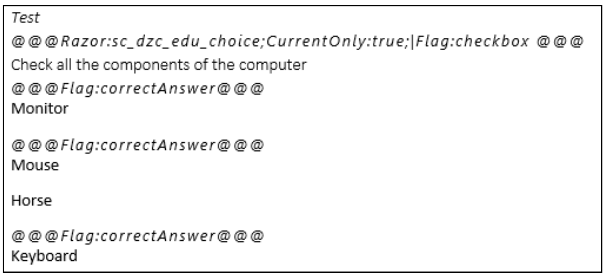
A specific feature of the EMMA tool is an initial drawing up of a learning content source in XML-compatible text editor (e.g., Microsoft Word), where all e-learning objects, which will constitute the executable learning content at the end of the production process, are defined – text, audio, video objects, test questions, navigations sequences, as well as visualisation rules. Audio and video objects may be stored in a dedicated folder and linked to the main .docx file. Table III shows an example of global metadata for learning content.

TABLE III

Learning Content of Global Metadata

Parameter	Value	Description
Razor	sc_main	Main learning content source processing Razor
Level	3	Creation of new pages according to the specified heading level; it defines the learning content structure
CurrentOnly	false	Defines if Razor relates to all learning content, or until next time, when Razor is defined in the source file
Theme	eLearncorm	Link to a folder, where learning content design elements reside
Flag	VISC;04;2	Additional attributes defining various functionalities and designs

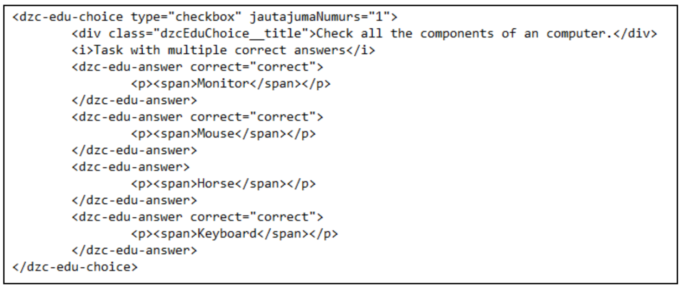
All e-learning objects in the mentioned file are defined by using a pseudocode. An example of the object description for "*choice*" type interaction is given in Fig. 2.



```
Test
@@@Razor:sc_dzc_edu_choice;CurrentOnly:true;|Flag:checkbox @@@
Check all the components of the computer
@@@Flag:correctAnswer@@@
Monitor

@@@Flag:correctAnswer@@@
Mouse

Horse

@@@Flag:correctAnswer@@@
Keyboard
```

Fig. 2. “*Choice*” type interaction pseudocode.

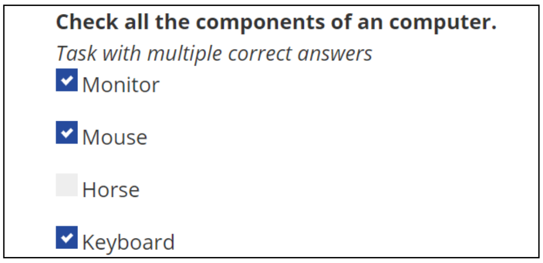
Further, Microsoft Word text with pseudocodes for the respective learning content can be converted to HTML5 (Fig. 3).



```
<dzc-edu-choice type="checkbox" jautajumaNumurs="1">
        <div class="dzcEduChoice__title">Check all the components of an computer.</div>
        <i>Task with multiple correct answers</i>
        <dzc-edu-answer correct="correct">
                <p><span>Monitor</span></p>
        </dzc-edu-answer>
        <dzc-edu-answer correct="correct">
                <p><span>Mouse</span></p>
        </dzc-edu-answer>
        <dzc-edu-answer>
                <p><span>Horse</span></p>
        </dzc-edu-answer>
        <dzc-edu-answer correct="correct">
                <p><span>Keyboard</span></p>
        </dzc-edu-answer>
</dzc-edu-choice>
```

Fig. 3. Generated HTML5 code.

The visual representation of the code execution in the LMS environment is also available, showing how the generated HTML code is translated in the browser as an interactive exercise (Fig. 4). The defined and generated learning content is fully operational in Moodle v3.8 LMS as SCORM 1.2 compliant SCO.



Check all the components of an computer.

Task with multiple correct answers

☑ Monitor

☑ Mouse

□ Horse

☑ Keyboard

Fig. 4. The result of learning content code generation in the LMS environment.

Research on the use of SCORM-compliant learning content in the Moodle environment revealed that some of elements of the data model related to the SCORM interactions (i.e., “*cmi.interactions.n.correct_responses._count*” and “*cmi.interactions.n.student_response*”) are not always applicable to all SCORM-compliant LMSs. If the intent is to save data about the learning progress within the SCORM learning content and use them in the next session, the only universally recognised element of the data model for SCORM to read and write these data is “*cmi.suspend_data*”.

The elements “*cmi.interactions.n.student_response*” and “*cmi.interactions.n.correct_responses._count*” appear to be write-only which means that a SCORM object can send learner’s answers only using this element of the data model. Unfortunately, Moodle does not use these SCORM capabilities. Moodle allows for the adoption of only one single percentage-assessed result from the test in one SCO. This means that in the SCORM learning content, interpreted by Moodle, all test questions must be placed in a single SCO since these objects cannot interact with each other to communicate the results.

V. Conclusion

The research resulted in the subset of SCORM requirements, which can be used to create learning content with the high level of reusability. The subset allowed for the definition of additional functional requirements for the learning content development tool EMMA developed and owned by a Latvian enterprise “Datorzinibu centrs”. They lead to the modifications in the specification of basic e-learning objects supported by the EMMA tool. The functional requirements for the EMMA tool are also extended to support not only the export of learning content into HTML, xAPI, and PDF format, but also into SCORM 1.2. After the implementation of changes, the tool was used for content production to validate interoperability aspects and allowed achieving the necessary interoperability.

Besides, the implementation of the defined SCORM requirements has revealed that there is a number of useful features of the SCORM 1.2 standard that are not supported by Moodle. Therefore, unless Moodle is to be considered one of LMSs with the best SCORM support, the interoperable content shall include quite a limited set of SCORM features.

Additionally, the collection of a wide range of SCORM-compliant learning content authoring tools and players has been created and analysed. The results of the analysis of so many tools have not been identified in other information sources. They can be used for a targeted selection of tools satisfying specific needs of institutions and specific users related to the creation and/or playing of learning content.

VI. Acknowledgment

The research leading to these results has been funded by the European Regional Development Fund project “Competence Centre of Information and Communication Technologies”, (1.2.1.1/18/A/003), individual research project 2.5 “Provision of e-learning object creation, migration and deployment interoperability in cloud infrastructure”.

References

[1] Global Industry Analysts, Inc., “E-learning – Global market trajectory & Analytics,” 2020. [Online]. Available: https://www.strategyr.com/market-report-e-learning-forecasts-global-industry-analysts-inc.asp [Accessed: Sep. 19, 2020].

[2] E. Duffin, “E-learning and digital education - Statistics & Facts,” Feb. 6, 2020. [Online]. Available: https://www.statista.com/topics/3115/e-learning-and-digital-education/ [Accessed: Sep. 19, 2020].

[3] Docebo, “E-Learning Trends 2019,” 2018. [Online]. Available: https://kometa.edu.pl/uploads/publication/634/550a_A_Docebo-E-Learning-Trends-2019.pdf?v2.8 [Accessed: Sep. 26, 2020].

[4] Technavio, “E-learning Market in Europe by Product and Geography - Forecast and Analysis 2020-2024,” May 2020. [Online]. Available: https://www.technavio.com/report/e-learning-market-in-europe-industry-analysis [Accessed: Sep. 19, 2020].

[5] N. Gilbert, "72 Essential LMS & eLearning Statistics: 2020 Market Share & Data Analysis," 2020. [Online]. Available: https://financesonline.com/25-essential-learning-management-system-e-learning-statistics-analysis-of-trends-data-and-market-share/#market [Accessed: Sep. 26, 2020].

[6] C. Ostyn, "In the Eye of the SCORM: An Introduction to SCORM 2004 for Content Developers," Mar. 2007. [Online]. Available: http://ucoalsumberdaya.com/uploads/files/Eye_Of_The_SCORM_draft.pdf [Accessed: Aug. 30, 2020].

[7] S. Sánchez-Alonso, M. G. López, and D. Frosch-Wilke, "E-learning Standards for Content Management," in *Content Management for E-Learn.*, 2011, pp. 131–156. https://doi.org/10.1007/978-1-4419-6959-0_8

[8] I. Kazanidis and M. Satratzemi, "A Framework for Automatic Construction of Reusable Adaptive Courses: The Case of ProPer SAT 2.0," in *Intelligent and Adaptive Educational-Learning Systems: Achievements and Trends*, 2013, pp. 213–238. https://doi.org/10.1007/978-3-642-30171-1_9

[9] Rustici Software, "A timeline and description of the eLearning standards," 2020, [Online]. Available: https://scorm.com/scorm-explained/business-of-scorm/scorm-versions/ [Accessed: Sep. 12, 2020].

[10] S. Balina, I. Arhipova, I. Meirane, and E. Salna, "Meta model of e-Learning materials development," in *Proceedings of the 16th International Conference on Enterprise Information Systems*, vol. 2, 2014, pp. 150–155. https://doi.org/10.5220/0004971401500155

[11] M. Bogdanovic and L. Stoimenov, "SCORM Player WP7: a Software Solution for Review and Presentation of the Learning Content on Mobile Devices," *TEM Journal*, vol. 1, no. 4, pp. 226–233, 2012.

[12] Thinking Cap, "Scorm, A Beginners Guide," Apr. 15, 2010. [Online]. Available: https://www.slideshare.net/ThinkingCamp/scorm-a-beginners-guide [Accessed: Sep. 12, 2020].

[13] Advanced Distributed Learning, "Technical Specification (Version 1.2)," Oct. 1, 2001. [Online]. Available: https://adlnet.gov/assets/uploads/SCORM_1_2_pdf.zip [Accessed: Sep. 11, 2020].

[14] C. Prpitsch and P. Veith, "Content and management standards: LOM, SCORM and Content Packaging," in *Handbook on Quality and Standardisation in E-Learning*, 2006, pp. 209–223. https://doi.org/10.1007/3-540-32788-6_15

[15] Advanced Distributed Learning, "SCORM Users Guide for Instructional Designers. SCORM 2004 4th Edition," Sep. 15, 2011. [Online]. Available: https://docplayer.net/14524462-Scorm-users-guide-for-instructional-designers-version-8.html [Accessed: Sep. 18, 2020].

[16] SCORM.com, "SCORM 1.2 Overview for Developers," 2020, [Online]. Available: https://scorm.com/scorm-explained/technical-scorm/scorm-12-overview-for-developers/ [Accessed: Sep. 3, 2020].

[17] B. S. Barker, "Adopting SCORM 1.2 Standards in a Courseware Production Environment," Int. J. on E-Learning, vol. 3, no. 3, pp. 21-24, 2004.

[18] Advanced Distributed Learning, "SCORM Users Guide for Programmers (SCORM 2004 4th Edition)," Sep. 15, 2011. [Online]. Available: https://tdmaryland.org/Resources/Documents/Program%20Documents/Feb Program_ADL/SCORM%20Users%20Guide%20for%20Programmers.pdf [Accessed: Sep. 19, 2020].

[19] R. T. Mason, *Interoperability Gap Challenges for Learning Object Repositories & Learning Management Systems*. USA, FL: Nova Southeastern University, 2011.

[20] H. P. Kong, W. K. H. Lim, L. Wang, and R. Gay, "SCMP: An E-Learning Content Migration and Standardization Approach (A Singaporean Perspective)," *J. of Distance Educ. Technol.*, vol. 4, no. 2, pp. 1–9, 2006. http://dx.doi.org/10.4018/jdet.2006040101

[21] E. Di Nitto, L. Mainetti, M. Monga, L. Sbattella, and R. Tedesco, "Supporting Interoperability and Reusability of Learning Objects: The Virtual Campus Approach," J. Educ. Technol. Soc., vol. 9, no. 2, pp. 3-50, 2006.

[22] R. P. Díaz Redondo, A. Fernández Vilas, and J. J. Pazos Arias, "Educateca: A Web 2.0 Approach to e-Learning with SCORM," in Cellary W., Estevez E. (eds) *Software Services for e-World*. I3E 2010. IFIP Advances in Information and Communication Technology, vol 341, Springer, Berlin, Heidelberg, 2010. https://doi.org/10.1007/978-3-642-16283-1_16

[23] Advanced Distributed Learning, "Sharable Content Object Reference Model (SCORM) Version 1.2: Conformance Requirements," Feb. 15, 2002. [Online]. Available: https://docplayer.net/46367355-Sharable-content-object-reference-model-scorm-tm-version-1-2-conformance-requirements-version-1-2-february-15-2002.html [Accessed: Sep. 10, 2020].

[24] Enovation, "Creating SCORM Content for Moodle," Oct. 27, 2015. [Online]. Available: https://enovationsolutions.co.uk/creating-scorm-content-for-moodle/ [Accessed: Sep. 22, 2020].

[25] Moodle, "Creating SCORM Content," Jul. 10, 2019. [Online]. Available: https://docs.moodle.org/39/en/Creating_SCORM_Content [Accessed: Sep. 13, 2020].

[26] Edutechwiki, "SCORM 1.2," Apr. 14, 2014. [Online]. Available: http://edutechwiki.unige.ch/en/SCORM_1.2 [Accessed: Sep. 5, 2020].

[27] T.-H. Wang and F. C.-I. Chang, "SCORM Compliant Courseware Authoring Tool for Supporting Pervasive Learning," in *Methods and Applications for Advancing Distance Education Technologies: International Issues and Solutions*, 2009, pp. 40–62. https://doi.org/10.4018/978-1-60566-342-5.ch004

Sintija Petrovica obtained the Doctoral degree of engineering sciences in the field of information technology, system analysis, modelling and design from Riga Technical University (RTU) in 2019. In 2011, she started to work as a Scientific Assistant at the Department of System Theory and Design at RTU (currently known as the Department of Artificial Intelligence and System Engineering). Since 2019, she has been working as a Researcher at the Department by participating in various e-learning projects. In addition, she also lectures both local and foreign students at RTU. Sintija's research interests include intelligent tutoring systems, affective computing, and game-based assessment.
Address: Department of Artificial Intelligence and System Engineering, Riga Technical University, 2 Daugavgrivas Str., Riga, LV-1048, Latvia.
Phone: +371 26569654
E-mail: sintija.petrovica@rtu.lv
ORCID iD: https://orcid.org/0000-0002-7670-638X

Alla Anohina-Naumeca is an Associate Professor at Riga Technical University with a nineteen-year experience of teaching in the field of computer science. She obtained the Doctoral degree in the field of information technology from Riga Technical University, Latvia in 2007. She defended the second PhD in the field of pedagogy on 3 May 2018 at the University of Latvia. Her main fields of interest are computer science, artificial intelligence, education, and educational software. She has more than 70 publications and the most part of them are indexed in Scopus and Web of Science. She has participated in more than 20 research projects of national and international project programmes.
Address: Department of Artificial Intelligence and System Engineering, Riga Technical University, 2 Daugavgrivas Str., Riga, LV-1048, Latvia.
Phone: +371 67089595
E-mail: alla.abohina-naumeca@rtu.lv
ORCID iD: https://orcid.org/0000-0001-7993-5842

Andris Kikans is the Chairman of the Board of Datorzinibu centrs, one of the leading regional learning providers. Andris has the Master of Business Administration degree and the Master degree in computer science. He has participated as a manager or researcher in many projects, involving the learning content development and learning platform implementation.
Address: Datorzinibu centrs Ltd., 41 Lacplesa Str., Riga, LV-1011.
Phone: +371 29191563
E-mail: andris.kikans@dzc.lv

APPENDICES

APPENDIX I. LIST OF THE SCORM AUTHORING TOOLS EXAMINED IN THE STUDY

SCORM authoring tool	**Link to the authoring tool**
Reload Editor	http://www.reload.ac.uk/editor.html
eXe Learning	https://exelearning.net/en/
Advanced SCORM Editor	https://advanced-scorm-editor.soft112.com/
Exerte	https://www.xerte.org.uk/index.php/en/
Easygenerator	https://www.easygenerator.com/en/
iSpring Free	https://www.ispringsolutions.com/ispring-free/download
Wondershare Quiz Creator	http://www.quiz-creator.com/
eLearning Authoring Tool	http://www.e-learningconsulting.com/software.html#authoring
ViewletQuiz	https://www.qarbon.com/presentation-software/viewletquiz/
GOMO	https://www.gomolearning.com/free-trial/
ActivePresenter	https://atomisystems.com/activepresenter/
Udutu Authoring Tool	https://www.udutu.com/elearning-authoring-tool/
Lectora Online	https://www.lectoraonline.com/
isEazy	https://iseazy.com/
CourseLab	http://www.courselab.com/view_doc.html?mode=doc&doc_id=5799960992579148560
Adobe Captivate	https://www.adobe.com/lv/products/captivate.html
Adobe Presenter	www.adobe.com/products/presenter/
ContentGenerator.net	http://www.contentgenerator.net/scorm/
docendo	http://www.docendo.org/home
Articulate 360 (Presenter un Quizmaker)	www.articulate.com
iSpring QuizMaker	https://www.ispringsolutions.com/ispring-quizmaker
Simple SCORM Packager (SSP)	https://www.jcasolutions.com/products/ssp/
Helius Presenter	https://helius-presenter.software.informer.com/
ToolBook	http://tb.sumtotalsystems.com/
Wondershare PPT2Flash (Professional Edition)	https://wondershare-ppt2flash-pro.soft112.com/
MOS Chorus LCMS	https://www.mindonsite.com/en/product/mos-solo/
ReadyGo Web Course Builder	http://www.readygo.com/
THESIS Rapid SCORM eLearning	http://thesis-rapid-scorm-elearning.hunterstone-inc.qarchive.org/
SCORM Developer Tools	www.e-learningconsulting.com/software.html#scorm
Coral Presenter	http://www.softsea.com/review/Coral-Presenter.html
Scormify	https://www.scormify.io/#!/
MS Learning Content Development System	https://www.microsoft.com/en-us/learning/lcds-tool.aspx
PowerQuizPoint Quiz Creator	https://www.digitalofficepro.com/powerpoint/quiz-creator.html

APPENDIX II. LIST OF THE SCORM PLAYERS EXAMINED IN THE STUDY

SCORM player	**Link to the SCORM player**
Moodle app SCORM Player	https://docs.moodle.org/37/en/Moodle_app_SCORM_player
RELOAD SCORM 1.2 Player	http://www.reload.ac.uk/new/scormplayer.html
SimpleAPI	https://github.com/skfriese/simple-scorm-api
SCORM 1.2 Test Wrap	https://www.jcasolutions.com/download/SCORM12WrapTest.zip
SCORM Proxy Player (SCORMPool)	http://www.scormpool.com/search/download.aspx
VSSCORM (A Very Simple SCORM Interface)	http://www.vsscorm.net/run-time-environment-rte/run-time-environment-downloads/
SCORM Player	https://www.softpedia.com/get/Multimedia/Video/Video-Players/SCORM-Player.shtml
Rustici Engine	http://rusticisoftware.com/products/rustici-engine/
SCORM Cloud	http://rusticisoftware.com/products/scorm-cloud/
Offline SCORM Player	https://www.paradisosolutions.com/blog/offline-scorm-player/

Scorm Preview Ninja	https://preview.coursesuite.ninja/
TinyLMS	https://www.randelshofer.ch/tinylms/
Simple SCORM LMS Adapter (SSLA)	https://www.jcasolutions.com/products/simple-scorm-lms-adapter/
Raccoon Gang SCORM Integrator xBlock	https://github.com/raccoongang/edx_xblock_scorm
Appsembler SCORM Xblock	https://github.com/appsembler/edx_xblock_scorm/
SCORM xblocx	https://github.com/mckinseyacademy/xblock-scorm