Integration of Pattern-Based Learning Management Systems with Immersive Learning Content

Michael Winterhagen[1][0000-0002-9487-3544], Dominic Heutelbeck[2][0000-00003-1395-2775], Benjamin Wallenborn[3][0000-0001-9234-5310], and Matthias L. Hemmje[1][0000-0001-8293-2802]

[1] Chair of Multimedia and Internet Applications (MMIA), Faculty of Mathematics and Computer Science, University of Hagen, Germany {michael.winterhagen, matthias.hemmje}@fernuni-hagen.de

[2] Research Institute for Telecommunication and Cooperation, Germany dheutelbeck@ftk.de

[3] University of Hagen, Center for Media and IT (ZMI), Germany benjamin.wallenborn@fernuni-hagen.de

Abstract. In our previous e-learning publications we introduced the concepts of *Course Authoring Tools* (CAT), and of *Didactical Structural Templates* (DST) which are a further development of the CAT. DSTs are defined as a possibility to describe the pedagogical structure of a course, a study program or an applied game in an abstract way. The idea of DSTs is based on the structure of IMS Learning Design (IMS-LD), which is a quasi-standard for modelling learning structures. We have shown what DSTs are useful for and that there is a need for an editor for DSTs which we already presented in combination with the *Didactical Structural Template Manager* (DSTM). In this paper we will focus on and go in deeper detail in implementing the same DST as a classic Learning Management (LMS) course, as a LMS course with applied gaming content and as a stand alone applied game. Therefore, these scenarios will show that it will be possible for learners to switch between different implementations of a specific DST and the learners having the same learning progress as if they had used just one of the implementations of this specific DST. One main contribution is to demonstrate the integration between, web-based, immersive Virtual Reality learning activities developed as native applications, and VR authoring tools by a gateway server. In addition, we present an RESTful API for sharing the DST and the used Competence Profiles to external tools.

Keywords: IMS Learning Design · IMS-LD · Didactical Structural Templates · Learning Tools Interoperability · LTI.

1 Introduction

Producing learning content for *Higher Education Institutes* (HEI) consumers is quite complex in *Learning Management Systems* (LMSs) for producers of learning content. The *Knowledge-Management Ecosystem Portal* (KM-EP) [6] is an Educational Portal which contains different management systems. A *Course Authoring Tool* (CAT) has been introduced in [16] within the KM-EP to reduce this complexity and make it simple to produce learning content in the way that the producers only have to fill in the necessary information and are able to concentrate in producing the learning content instead of configuring the learning content within the LMS.

1.1 Motivation, Problem Statement and Approach

The so-called *Didactical Structural Templates* (DST) have been introduced in [19] and extended in [17]. As described, the DSTs are based on the *IMS Learning Design* (IMS-LD) [8] and represent the pedagogical structure of a course and cannot only be used as pedagogical structure for creating courses. In fact, the DSTs can also be used as a pedagogical structure for a hybrid environment existing of a "classical" course with integrated applied gaming content just like a pedagogical structure for applied game which can be a web-based computer game or a Virtual Reality (VR)/ Augmented Reality (AR) based game. Therefore, one DST can have different implementations.

The advantage of this approach is, that learners will be able to switch between different implementations of one DST whenever they want to and they have got the same learning progress as if they had used only one specific implementation of this DST. This means if learners like gaming, they can use the applied gaming implementation to work on the learning content. If it is easier for the learners to answer the self-tests or the final test – to stay in the exemplary stated pedagogical structure of a course – as e.g. multiple-choice quizzes, they can switch to a course within an LMS to answer the questions.

Within a wide rage of domains, realizing learning activities as immersive experiences, e.g., in VR, may significantly improve the learners experience and success [1]. While tools like the WebXR Device API [15] have the potential to close the gap between traditional browser-based learning and content played out via dedicated VR Hardware. In industrial VR learning environments, specialized hardware-based controllers may be necessary for the appropriate simulation of machine operation, and the virtual environments are of high fidelity, derived from engineering CAD systems. Complex environments, such as models of complete factories are very demanding with regards to graphics capabilities of the hardware. Thus, learning activities and self tests in VR often require the use of native application running on a capable machine, connected to specialized controllers. While CATs are often also able to support the authoring of web-based content, immersive applications, both native and WebXR-based, require specialized authoring tools which are able to handle the spatial and graphical content and complex interfaces. In practice, such authoring tools are built on top of game development tools and platforms, such as the Unity [14] or Unreal [5] engine.

Such immersive activities are potentially embedded within a DST alongside traditional web-based content. The integration of this type of content into the authoring process, deployment and execution of a course instance implementing a DST rises a number of technical and architectural challenges.

During web-based learning in a traditional LMS, the use of external web-based tools is well supported by the *Learning Tools Interoperability* (LTI) [10] Standard by a number of modular APIs and protocols. This allows to start activities implemented using WebGL or WebXR directly. However the reliance of LTI on OAuth [13], *Transport Layer Security* (TLS) [7], and forwarding between tool server and LMS server for launch requests does prohibit directly launching into a native application installed on the users computer, while retaining authentication and course context information. E.g., launch URLs have to be public TLS secured web-resources. A locally running tool cannot fulfill the TLS requirements running on 'localhost'.

To summarize the remainder of this paper addressing our five research questions, which we will require to work on shown below:

1. How can a DST representation be applied to a "classical" course production in an LMS?
2. How is the responsibility of content creation distributed between the different content authoring tools.
3. How can we share and cross-reference content between the different tools and execution environments.
4. Can LTI-style launch requests be realized in order to launch into native applications while retaining the learners authentication and course context?
5. How can native external tools send back results to the LMS?

1.2 Methodology

As the basis of our research methodology, the multi-methodological framework of Nunamaker and Chen [12] is used for the structured research and development of information systems. The framework is divided into four phases supporting different methodological strategies: Observation, Theory Building, System Development, and Experimentation. To achieve our research goal to answer our research questions, the methodological phases can be executed repeatedly in any order. It is also possible to return to previous phases.

2 Concepts and Technologies

LTI is introduced in [11]. LTI supports a connection between LMSs and external applications like gaming platforms. An applied game prototype based on the *Unity Game Engine* (Unity) [14] has been introduced in [18]. This prototype is later used as an example for implementing an LTI connection. However, this only works for web-applications and not for desktop-applications. To enable this also for desktop applications, the solution presented in [18] has to be extended.

3 Conceptual Work

After describing the problems and research questions in chapter 1.1 we want to present our conceptual work in this chapter. To realize the DST as an applied game we have different challenges we have to work on, to make this option work:

1. LTI can only be used for web-based applications, not for stand alone or desktop-applications. How can LTI be used for our scenario?
2. How can the applied game access the DST?

3.1 LTI Launch Requests and Services for Immersive Native Applications

In this section we present our architecture (see figure 1) and implementation of a technical solution for the above described challenges arising from the need for *Immersive Native Applications* (INAs) in order to support immersive VR learning content. Overall, the idea is to use a so-called *LTI Gateway* to mediate between a) the LMS implementing LTI, b) the VR-specific authoring tools, and c) the VR learning tool installed on the users computer. The goal is to support standard LTI 1.3 [10] launch requests to trigger learning activities using the local INA and to allow it to use LTI Advantage [11] services offered by the platform (LMS), e.g. Assignment and Grade Services 2.0 [9].

In LTI 1.3 a platform and an external tool have to be carefully paired and the tool and platforms have to be authorized by the respective administrators. In this process, public keys and several URLs have to be exchanged and unique IDs for tool and platform have to be provisioned. Generally the URLs have to be mutually reachable by the systems hosting either component and it is mandatory for both sides to have all connections secured by TLS supported by the matching certificates.

In order to use the INA, it has to be installed locally on the machine used by the learner. While in special circumstances, this may be a dedicated machine where it is possible to assign a DNS entry and to obtain a TLS certificate. Then, the INA could directly support the LTI protocols by running matching background services. However, in practice learners will not use a singe machine shared among them. Thus, even in a learning environment with multiple machines that could fulfill these conditions, in the LTI domain model each machine would constitute a different external tool which implies that in a course the same learning activity on different machines would be listed separately. Further more, it is likely that a course is not consumed by a predetermined number of local learners, but by a variable number of remote learners with machines in uncontrolled environments and behind NAT routers. Thus, a direct provisioning of local installations of the INA as LTI tools is not feasible.

In order to use other LTI Advantage services, external tools acquire OAuth 2.0 [13] access tokens. While the OAuth 2.0 standard technically allows for INAs to obtain such tokens directly, it is not guaranteed, that the matching OAuth flows are supported by all LMS. However the flows for web-based applications to obtain the tokens is mandatory for the LTI standards.

To support both launch request and and LTI Advantage services, a dedicated web-based service, the LTI Gateway is introduced. The LTI Gateway itself acts as a single point of entry and an LTI tool for the platform and is able to relay the launch request to an INA and service calls from the INA to the platform. Thus, pairing between tool and platform only happens once for the gateway and not for each local installation of the INA.

The gateway can be referenced in a course like any other external LTI tool. When a user decides to access the learning activity and clicks on the matching link provided by the platform, a standard LTI Launch Request to the gateway is triggered. In the process of the launch request protocol, the gateway obtains information about the user and the context of the launch via different claims stored in the *JSON Web Token* (JWT) [4] used in the launch. In addition, the gateway obtains an access token to use the platforms LTI Advantage services. The course author may have decided to add additional custom parameters (string key-value pairs) to the launch, which are also made available to the tool in the matching JWT claim.

Given, that users may have different sections and activities of a course open at the same time or that users may, often unintentionally, click a launch link multiple times the gateway has both to disambiguate launch requests based on the user and context information.

Thus, the gateway is a stateful service which upon receiving a launch request and obtaining the information creates a gateway session identified by the tuple of user identifier and LTI context. Each session has a predefined, decided by the gateway administrator, *time to live* (TTL) and a randomly generated Base64 encoded 128-bit *Universally Unique Identifier* (UUID), acting as a session key. For this session, the triple, session key, LTI Advantage access token, and JWT token are persistent in RAM for the TTL. Before creating a new session for a launch request, it is checked if another session for the given triple already exists, which is used if present.

After creating, or loading the session, the learner is presented with a TLS secured web page containing information on the origin of the launch and instructions on how to obtain the INA, if the user still needs to install it. The user is presented with a launch link to start the INA. The launch link is structured as follows: `[CUSTOM PROTOCOL]:[TOOL_NAME]?sessionKey=[SESSION_KEY]`, e.g., `i2l:virtualFactory?sessionKey=jhSLqIopUf_ZiP86n6VDaA`. At install time, the INA registers itself as the handler for the custom protocol (e.g., adding the respective registry entries for Windows). Thus, when clicking the link, the web browser will launch the INA and provide the URL as a command line argument. In this way, the INA is able to obtain the session key. Compared to a direct authorization of the INA and its user via OAuth this opens an attack vector, where on a compromised machine the launch may be intercepted and the session key may be obtained by a malicious application. This however is only possible as a local attack and the key is always encrypted in transit over the network.

We consider this to be an acceptable remaining risk. This requires local code execution privileges which imply further extensive attack potential.

Given the session key, the INA can retrieve the session data from the gateway by accessing GET: <`URL`>`/launchRequest/`{`sessionKey`}, returning a JSON object containing user information, LTI resource link, line item, and context and custom data provided by the course author.

The INA can now start the learning activity according to this information. For each LTI Advance service the gateway also provides matching proxy services which accept the session key as authentication, such as results or scoring services. Whenever the INA accesses the proxy services with the session key, internally the request is mapped to a matching LTI request to the platform, using the stored access token, which is never shared with the INA.

3.2 VR Authoring and Resource Server

Figure 1 gives a rough overview of the interaction of the different parts described in the previous sections.

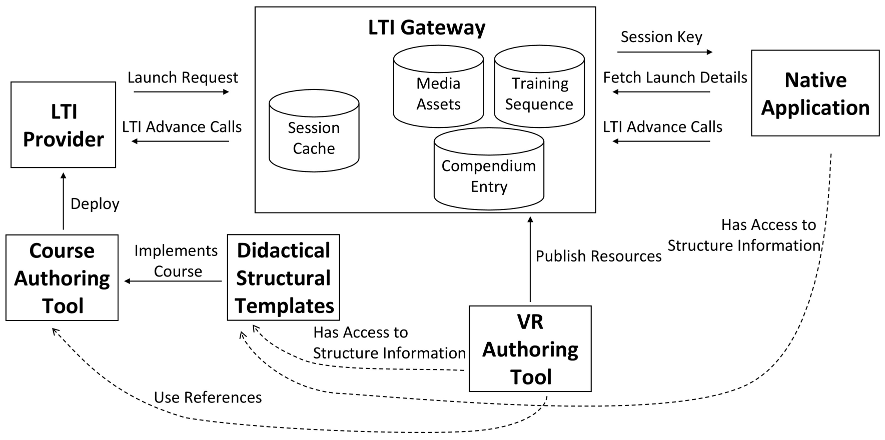


Fig. 1. LTI Gateway and VR Authoring Tools

Within the VR Authoring Tool, the producers can define the Training Sequences, Compendium Entries and Media Assets which are published on a server which holds this content. The meaning of this elements are as follows:

The *Media Assets* can be texts or pictures and can be used by Compendium Entries.

The *Compendium Entries* can be understood as elements of knowledge transfer. They consist of contents in a specific order. Each content is composed of e.g.

texts and pictures which are stored as Media Assets. The *Compendium Entries* e.g. deliver information about the environment of the game or the Training Sequences. To access the Media Assets, the Compendium Entry has to connect to the content server and download it from there.

At last, the *Training Sequences* can be understood as learning elements of a learning path. This Training Sequences consist of so-called *Tasks* which are in a specific order and have to be done in the defined order. Each Task contains so-called *Objectives* which have no specific order. Each Objective will be displayed in a menu in the game. To complete a task, each Objective has to be done completely.

Having described the *Training Sequences* and its structure, we can define a mapping between the IMS-LD and an applied game using the *Training Sequences* as shown in table 1.

Table 1. Mapping IMS-LD - applied game

IMS-LD element	applied game element
method	applied game
play	levels
act	Training Sequence
activity structure	Task
activity	Objective

3.3 Delivering the Structure of a Didactical Structural Template to an External Application

In this section we present our technical solution for the above described challenge 2. We support two kinds of delivering the structure of an DST to an external application:

1. The *Didactical Structural Template Manager* (DSTM) provides the possibility to export an DST as XML-file which meets the IMS-LD specification. Therefore, the DST could be exported in this format and can be used to design the structure of the game.
2. The KM-EP provides a Learning Design API which will be described in short in the following. The external application can access this Learning Design API to get all information about the needed DST.

The Learning Design API is defined as a *Representational State Transfer* (REST) [3] with the following endpoints:

– GET: <URL>/webservice/learningdesign
  This endpoint returns a list of all available DSTs.
– GET: <URL>/webservice/learningdesign/{id}
  This endpoint returns the structure of a specific DST.

The return value of each endpoint is made in *JavaScript Object Notation* (JSON) [2]. The specific return value's format will not be explained in detail here. With this both kinds of delivering the DST to an external application, it is possible for e.g. an applied game to access the needed DST.

4 Prototypical Implementation

After describing the technical requirements and the technical prerequisites, we are now ready to show, how it is possible to implement a DST as

1. a normal Moodle course,
2. a gamified Moodle course, and
3. an applied game.

This will be done within the following subsections.

4.1 Implementing a Didactical Structural Template as a Moodle Course

After defining an DST at first we want to implement this into a Moodle course. To do so, we have to go to the DSTM within the KM-EP and press on the *install* button next to the DST we want to implement.

After pressing the *install* button, the DSTM will implement the selected DST as Moodle course according to the mapping we presented in [17] with default values for each learning element. Therefore, we now have an empty Moodle course created which we have to fill with the needed learning content.

To fill this empty Moodle course with learning content, we can now use the CAT within the KM-EP to edit every learning element of the created Moodle course.

After saving the changed learning element, the changes will directly be transferred to Moodle. After the complete course has been filled with the needed learning content, the course is ready to be published and used by learners.

4.2 Implementing a Didactical Structural Template as a Gamified Moodle Course

As second option we want to implement the same DST of 4.1 as a Moodle course but with an applied game.

Before we implement the course, there has to be implemented an applied game with an LTI interface. For our example, we use a web-based applied game implemented with the *Unity Engine* (Unity) [14].

We already described in detail in [18] how the LTI connection between Moodle and the applied game works. We will show exemplary, how the implemented applied game can be included within an Moodle course.

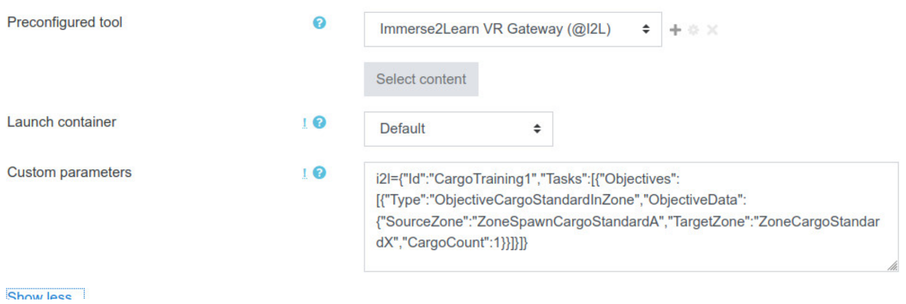
At first the URL of the created applied game has to be added as software into the KM-EP.After the applied game is known within the KM-EP, it can be referenced by a Moodle course. Therefore, an learning element has to be defined as asset which references the applied game with help of the CAT.

When a learning element with an underlying applied game is selected by the learners, the applied game will start and the learning result will be transferred back to the LMS via LTI.

4.3 Implementing a Didactical Structural Template as an Applied Game

As third option we want to implement the same DST of 4.1 as a stand alone applied game. In our scenario, this game will be an VR/AR game.

Prototypical Implementation of LTI connections for Stand Alone Applications To make it possible to call the LTI Gateway mentioned in 3.1, we have to define an LTI connection in Moodle (see figure 2).



Preconfigured tool	Immerse2Learn VR Gateway (@I2L)
Launch container	Default
Custom parameters	i2l={"Id":"CargoTraining1","Tasks":[{"Objectives":[{"Type":"ObjectiveCargoStandardInZone","ObjectiveData":{"SourceZone":"ZoneSpawnCargoStandardA","TargetZone":"ZoneCargoStandardX","CargoCount":1}}]}]}

Fig. 2. Define LTI connection in Moodle.

For our purpose we need at least version 1.3 of the LTI specification. After the LTI connection for the LTI Gateway has been defined within Moodle for a specific learning element, a click on the corresponding learning element will open a website which provides a link to the applied game.

After clicking on the provided link, an applied game will be downloaded and started. This applied game will contain the LTI connection to return learning results to Moodle.

5 Conclusions

In this paper we have shown how to implement a given DST

- as a normal Moodle course,
- as a gamified Moodle course, and
- as an applied (VR) game without Moodle or another LMS.

We also have extended the LTI connection, which until now only could be used for web-based applications, for stand alone applications via an LTI Gateway. We used this LTI Gateway for an applied game which implemented an DST.

To give an external application access to the pedagogical structure of an DST we offered two options:

1. exporting the pedagogical structure of an DST via the extended IMD-LD specification and
2. providing a REST-API which provides the pedagogical structure of an DST in JSON format.

If we have two different DSTs, namely *DST A* and *DST B*, the learners learning progress can be further developed, when the learners switch between different implementations of *DST A*. If they switch into an implementation on *DST B*, they will have anoter, separate learning progress apart from *DST A*.

5.1 Future Work

Future work will be a further formal evaluation of the DSTs and the DSTM, especially in the way to use the DST as pedagogical structure for an applied game without connecting it to a Moodle course. Also a formal further evaluation will be based on learners switching between different implementations of an DST having the same learning progress.

The process of creating a Moodle course by installing it from an DST is not as good as it could be. To improve the installation process, it would be desirable to have a wizard, so that the learning content can already be filled in while installing the DST as a course.

We also have to extend the possibility to connect to applied games via LTI for desktop-applications.

Acknowledgements

This publication has been produced in the context of the RAGE and Immerse2Learn projects. The projects have received funding from the European Union's Horizon 2020 Research and Innovation Action under grant agreement No 644187 and from European funds for regional development (EFRE). However, this paper reflects only the authors' views and the European Commission is not responsible for any use that may be made of the information it contains.

References

1. Allcoat, D., von Mühlen, A.: Learning in virtual reality: Effects on performance, emotion and engagement. Research in Learning Technology **26** (2018). https://doi.org/https://doi.org/10.25304/rlt.v26.2140, `https://journal.alt.ac.uk/index.php/rlt/article/view/2140`

2. Crockford, D.: The javascript object notation (json) data interchange format (2017), https://tools.ietf.org/pdf/rfc8259.pdf
3. Fielding, R.T.: Architectural Styles and the Design of Network-based Software Architectures. Ph.D. thesis (2000), https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation_2up.pdf
4. Force, I.E.T.: Rfc 7519 - json web token (jwt) (2015), https://tools.ietf.org/html/rfc7519
5. Games, E.: The most powerful real-time 3d creation platform - unreal engine (2020), https://www.unrealengine.com/en-US/
6. Globit-GmbH: Ecosystem portal, https://www.globit.com/#ep
7. IETF: The tls protocol (1999), https://tools.ietf.org/html/rfc2246
8. I.G.L.-Consortium: Learning design specification (2003), https://www.imsglobal.org/learningdesign/index.html
9. I.G.L.-Consortium: Ims learning tools interoperability (lti) assignment and grade services specification (2019), https://www.imsglobal.org/spec/lti-ags/v2p0/
10. I.G.L.-Consortium: Learning tools interoperability core specification (2019), http://www.imsglobal.org/spec/lti/v1p3/
11. I.G.L.-Consortium: Lti v1.3 and lti advantage (2019), https://www.imsglobal.org/activity/learning-tools-interoperability
12. Nunamaker, J.F., Chen, M., Purdin, T.M.D.: System development in information systems researchs (1990), http://gkmc.utah.edu/7910F/papers/JMIS%20systems%20development%20in%20IS%20research.pdf
13. OAuth: Oauth 2.0 (2012), https://oauth.net/2/
14. Unity: Unity technologies, https://unity3d.com/de
15. W3C: Webxr device api (2019), https://www.w3.org/TR/webxr/
16. Wallenborn, B.: Entwicklung einer innovativen Autorenumgebung für die universitäre Fernlehre. Ph.D. thesis, FernUniversität Hagen (2018), https://ub-deposit.fernuni-hagen.de/receive/mir_mods_00001428
17. Winterhagen, M., Hoang, M.D., Lersch, H., Fischman, F., Then, M., Wallenborn, B., Heutelbeck, D., Fuchs, M., Hemmje, M.: Supporting structural templates for multiple learning systems with standardized qualifications. In: EDULEARN20 Proceedings (2020)
18. Winterhagen, M., Salman, M., Then, M., Wallenborn, B., Neuber, T., Heutelbeck, D., Fuchs, M., Hemmje, M.: Lti-connections between learning management systems and gaming platforms. Journal of Information Technology Research **13** (2020)
19. Winterhagen, M., Then, M., Wallenbrn, B., Hemmje, M.: Towards structural templates for learning management systems taking into account standardized qualifications. Tech. rep., FernUniversität Hagen (2019), https://www.researchgate.net/publication/340601087_Towards_Structural_Templates_for_Learning_Management_Systems_taking_into_account_Standardized_Qualifications