QDITA and LAMS Integration

NIKOLA GUDURIC
Irvas international d.o.o, nikola.guduric@irvas.rs

LJUBISA JOVEV
Irvas international d.o.o., ljubisa.jovev@irvas.rs

BOBAN NIKOLIC
Irvas international d.o.o., boban.nikolic@irvas.rs

Abstract: *This paper presents one possible approach in QDITA (authoring system for learning material preparation) and LAMS (Learning Activity Management System) integration. The paper focus is on technical aspects of the system integration which is based on QDITA enhancement. New functionality in QDITA (LAMS package creation) is implemented, which provides easy transfer of DITA content to LAMS sequences.*

Keywords: *E-Learning, Distance learning, DITA, LAMS*

1. INTRODUCTION

The essence of the QDITA [2] approach is the concept of structural writing of granular content, based on the DITA framework philosophy. The main features and concepts provided by DITA standard [1] are reusability, content model, content management and publishing. More about these features, their importance and the way they are realized in QDITA can be read in our previous paper named “QDITA: AN ECO-SYSTEM FOR ADAPTIVE E-LEARNING” [2]. Furthermore, IMS-LD standard [3] supports the use of a wide range of pedagogies in online learning.

The language was originally developed at the Open University of the Netherlands (OUNL), after extensive examination and comparison of a wide range of pedagogical approaches and their associated learning activities, and several iterations of the developing language to obtain a good balance between generality and pedagogic expressiveness.

As DITA learning specialization does not support Learning Activities, it cannot meet requirements to organize online lessons in more interactive way. However, LAMS [4] is a system that enables and implements these requirements.

Main topic of this paper is technical solution of QDITA and LAMS integration, having in mind importance of learning activities in E-Learning

This paper presents a solution of QDITA and LAMS integration, based on introduction of Learning Activity topic and topic reference in QDITA. This is the basic idea for the implementation of LAMS compatible package format.

2. LEARNING ACTIVITY

As the pedagogy of the twentieth century was marked with the cognition that learning is not a simple process "of receiving knowledge" but an active process in which the student contextualizes, improves his knowledge, and masters learning strategies, there was a need to apply these pedagogical principles in e-learning. At the same time, an active form of education in which students interact with learning content using their newly and previously acquired knowledge is favored.

E-Learning elements that provide student’s activities and improve communication between students and professors and students themselves are called Learning Activities. Some of the most common learning activities are assessment, forum and chat, media transfer, shared documents, surveys etc.

These important elements of learning process are not supported by DITA standard. DITA provides structural and reusable content that can be published in different formats, but these learning activities are not part of the standard.

3. LAMS

LAMS (Learning Activity Management System) is an open source Learning Design system for designing, managing and delivering online collaborative learning activities. These activities can include a range of individual tasks, small group work and whole class activities based on both content and collaboration. [1]

[1] [1] This paper provides some results obrained by the project with code III44006 financed by the Ministry of Education, Science and Technological Development of Rebublic of Serbia

The development of LAMS was based on the conceptual model of Learning Design, and on parts of the IMS LD specification. However, LAMS encountered a range of practical problems with the specification during implementation, so it was not possible to implement the specification in full in the original LAMS development. For this reason, LAMS is often referred to as being IMS LD "inspired" rather than a reference implementation.

LAMS introduces wide range of learning activities. Noticeboard, Assessment, Forum, Q&A, Survey etc, are some of them. Primary use of Noticeboard activity is to show learning content and that is why it deserves special attention.

More LAMS activities are associated in one LAMS sequence. Since a teaching unit may be represented as a set of learning activities, each LAMS sequence corresponds to one teaching unit. In the DITA domain this corresponds to one DITA Learning Map.

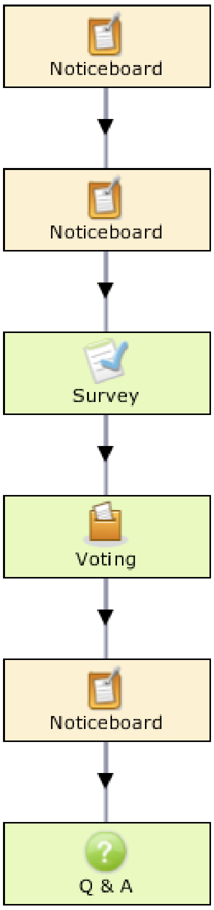


Figure 1: LAMS sequence

Each LAMS sequence can be exported from LAMS. Exported LAMS sequence is actually a .zip file named LAMS package. Such LAMS package can be imported back to the same LAMS installation it was exported from, or to another one. This fact was the starting point of LAMS and QDITA integration.

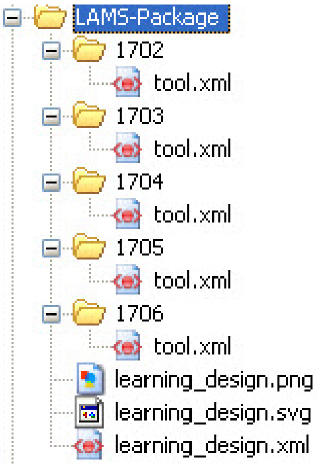
4. QDITA AND LAMS INTEGRATION

DITA content may be published in many formats such as pdf, doc, e-pub, html etc, but does not support publishing in LAMS package format. For the integration purpose the main task was to extend QDITA's publishing to create LAMS package that has reference to DITA OT [6] publishing output.

In order to answer this task, it was needed to become familiar with the internal structure of the LAMS package, and understand it.

4.1 LAMS package internal structure

As mentioned before, LAMS package is actually a ZIP file.



```
LAMS-Package
    1702
        tool.xml
    1703
        tool.xml
    1704
        tool.xml
    1705
        tool.xml
    1706
        tool.xml
    learning_design.png
    learning_design.svg
    learning_design.xml
```

Figure 2: Structure of LAMS package

It is composed of *learning_design.xml* manifest file and *tool.xml* for each learning activity from sequence. Manifest file (*learning_design.xml*) is similar to IMS LD standard. Structure of manifest file is shown in Fig. 3.

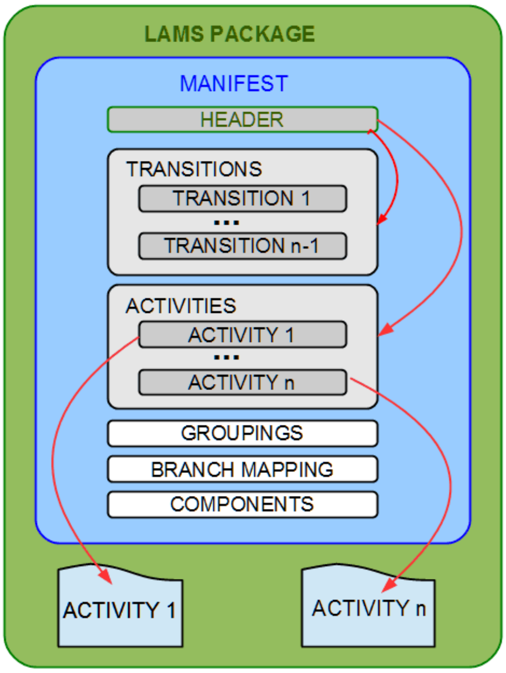


Figure 3: LAMS package

In activities part of a manifest file, there is an activity node for each learning activity from sequence. Each activity node has a reference to *tool.xml* file where that activity information is stored. That tool.xml file is stored in a folder named by the corresponding activity's id in the manifest. The order of activities in a sequence is defined in the transitions part of the manifest file.

The structure of tool.xml file depends on the activity type.

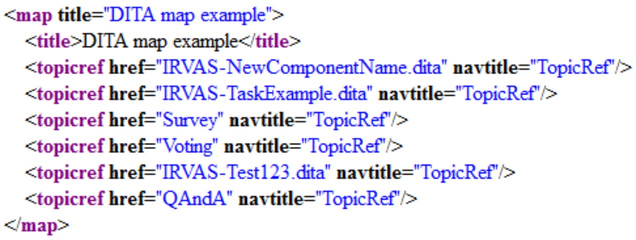
4.2 LAMS sequence and DITA map

There are two issues in QDITA and LAMS integration or creation DITA package from DITA map. The first issue is how to define LAMS activities which are not supported by DITA standard in DITA map. The second issue is, of course, how to create LAMS sequence from DITA native content using DITA publishing.

4.3 DITA customization

There are two alternatives to the first issue solution. The first alternative is DITA specialization [5]. DITA specialization allows us to define new kinds of information, new DITA types, extending the existing ones. This means that we would have to extend existing DITA's DTDs and XSDs for each LAMS activity type. This solution wouldn't be that simple, so we found another one.

Second solution, DITA customization, doesn't include any DITA enhancements. To define any LAMS activity in a DITA map, "topicref" element is used, as for any other DITA type, but with specific value of "href" attribute.



```
<map title="DITA map example">
   <title>DITA map example</title>
   <topicref href="IRVAS-NewComponentName.dita" navtitle="TopicRef"/>
   <topicref href="IRVAS-TaskExample.dita" navtitle="TopicRef"/>
   <topicref href="Survey" navtitle="TopicRef"/>
   <topicref href="Voting" navtitle="TopicRef"/>
   <topicref href="IRVAS-Test123.dita" navtitle="TopicRef"/>
   <topicref href="QAndA" navtitle="TopicRef"/>
</map>
```

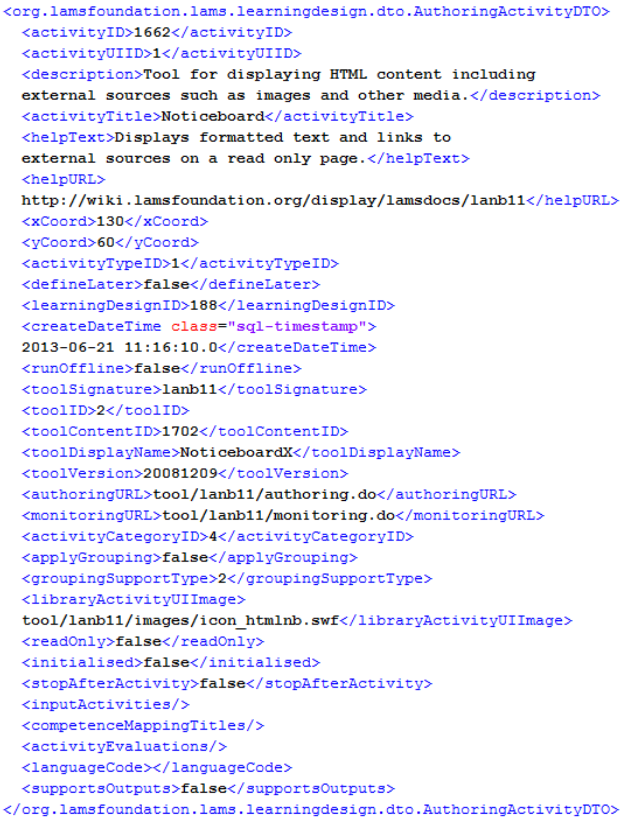
Figure 4: DITA map example

DITA map from Fig. 4 corresponds to the LAMS sequence from Fig. 1. From these figures it can be noticed that for each Noticeboard in DITA map topicref reference some DITA topic element, while other LAMS activities has href value corresponding to the LAMS activity title.

This fact is important in resolving the second issue, the creation of LAMS package from DITA map.

4.4 Creating LAMS package from QDITA

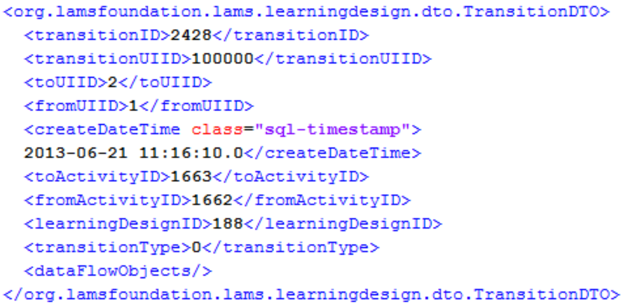
Each topicref element in DITA map will be one activity in LAMS sequence. The creation of LAMS package includes creation of manifest file and *tool.xml* file for each activity. For each activity from DITA map, activities part in the manifest file needs to be updated, and some information about activity recorded here. Some information about activity being stored here are activity type, title, description and name of the folder where the corresponding *tool.xml* file is stored. Part of manifest file that represents one Noticeboard activity is shown in Fig. 5.



```
<org.lamsfoundation.lams.learningdesign.dto.AuthoringActivityDTO>
  <activityID>1662</activityID>
  <activityUIID>1</activityUIID>
  <description>Tool for displaying HTML content including
  external sources such as images and other media.</description>
  <activityTitle>Noticeboard</activityTitle>
  <helpText>Displays formatted text and links to
  external sources on a read only page.</helpText>
  <helpURL>
  http://wiki.lamsfoundation.org/display/lamsdocs/lanb11</helpURL>
  <xCoord>130</xCoord>
  <yCoord>60</yCoord>
  <activityTypeID>1</activityTypeID>
  <defineLater>false</defineLater>
  <learningDesignID>188</learningDesignID>
  <createDateTime class="sql-timestamp">
  2013-06-21 11:16:10.0</createDateTime>
  <runOffline>false</runOffline>
  <toolSignature>lanb11</toolSignature>
  <toolID>2</toolID>
  <toolContentID>1702</toolContentID>
  <toolDisplayName>NoticeboardX</toolDisplayName>
  <toolVersion>20081209</toolVersion>
  <authoringURL>tool/lanb11/authoring.do</authoringURL>
  <monitoringURL>tool/lanb11/monitoring.do</monitoringURL>
  <activityCategoryID>4</activityCategoryID>
  <applyGrouping>false</applyGrouping>
  <groupingSupportType>2</groupingSupportType>
  <libraryActivityUIImage>
  tool/lanb11/images/icon_htmlnb.swf</libraryActivityUIImage>
  <readOnly>false</readOnly>
  <initialised>false</initialised>
  <stopAfterActivity>false</stopAfterActivity>
  <inputActivities/>
  <competenceMappingTitles/>
  <activityEvaluations/>
  <languageCode></languageCode>
  <supportsOutputs>false</supportsOutputs>
</org.lamsfoundation.lams.learningdesign.dto.AuthoringActivityDTO>
```

Figure 5: Activity in Manifest file

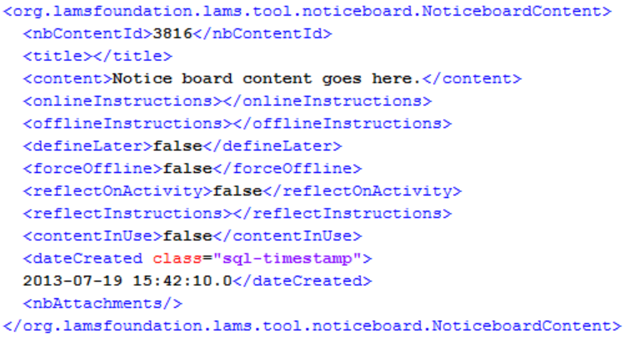
In transitions part of a manifest file each relation between two activities needs to be stored. Each transition has an "activity from" id and an "activity to" id, transition type, transition id, etc. A part of a manifest file that represents one transition is shown in Fig. 6.



```
<org.lamsfoundation.lams.learningdesign.dto.TransitionDTO>
  <transitionID>2428</transitionID>
  <transitionUIID>100000</transitionUIID>
  <toUIID>2</toUIID>
  <fromUIID>1</fromUIID>
  <createDateTime class="sql-timestamp">
  2013-06-21 11:16:10.0</createDateTime>
  <toActivityID>1663</toActivityID>
  <fromActivityID>1662</fromActivityID>
  <learningDesignID>188</learningDesignID>
  <transitionType>0</transitionType>
  <dataFlowObjects/>
</org.lamsfoundation.lams.learningdesign.dto.TransitionDTO>
```

Figure 6: Transition in manifest file

As it was mentioned above in the text, the content of each *tool.xml* file depends on the activity type. Information describing LAMS activity content is stored here. For Noticeboard it is title, content and some parameters like creation date, etc. For survey it is instructions, list of questions, possible answers etc. At the moment, from QDITA application it is possible to add only default LAMS activities, except Noticeboard. It means that default values of the tool.xml file won't be changed for any LAMS activity except for Noticeboard. For this reason we will keep the attention only to the Noticeboard's *tool.xml* file. Its code is shown in Fig. 7.



```
<org.lamsfoundation.lams.tool.noticeboard.NoticeboardContent>
  <nbContentId>3816</nbContentId>
  <title></title>
  <content>Notice board content goes here.</content>
  <onlineInstructions></onlineInstructions>
  <offlineInstructions></offlineInstructions>
  <defineLater>false</defineLater>
  <forceOffline>false</forceOffline>
  <reflectOnActivity>false</reflectOnActivity>
  <reflectInstructions></reflectInstructions>
  <contentInUse>false</contentInUse>
  <dateCreated class="sql-timestamp">
  2013-07-19 15:42:10.0</dateCreated>
  <nbAttachments/>
</org.lamsfoundation.lams.tool.noticeboard.NoticeboardContent>
```

Figure 7: Content of the Noticeboard's tool.xml file

To update Noticeboard's content we had to update <content> tag in the tool.xml file. Content of Noticeboard is determined by content of corresponding DITA topic. Noticeboard's content should be a HTML preview of the corresponding DITA topic. Therefore,

In order to show DITA topic's HTML preview in Noticeboard content, a DITA map was published as HTML. In the Noticeboard's *tool.xml* file, the <iframe> element that shows corresponding HTML page needs to be added into the content element. Updated value of <content> tag is showed below.

```
<content>
   <div>
      <iframe height="850" src="http://dita.metropolitan.ac.rs:8880/qdita-
         temp/publish/IRVAS/outxhtml_1372239655344/IRVAS-
                        LContentTask.html" width="100%">
      </iframe>
   </div>
</content>
```

4.5 LAMS package creation process

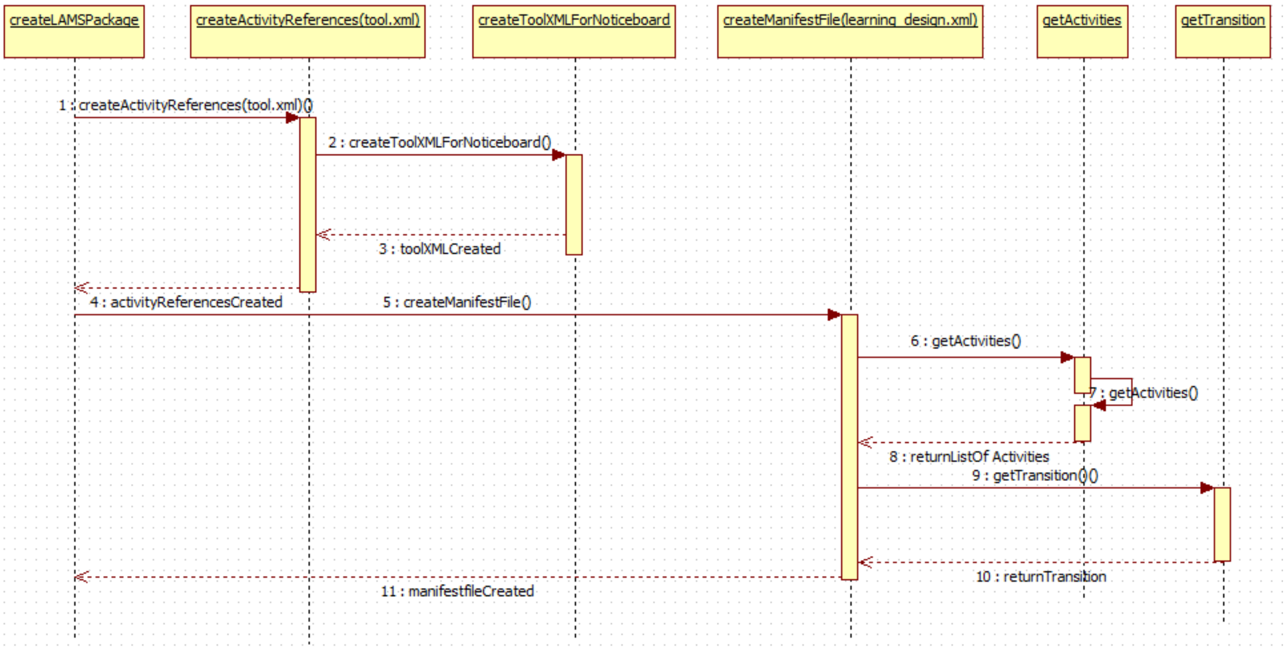

createLAMSPackage
createActivityReferences(tool.xml)
createToolXMLForNoticeboard
createManifestFile(learning_design.xml)
getActivities
getTransition
1 : createActivityReferences(tool.xml)()
2 : createToolXMLForNoticeboard()
3 : toolXMLCreated
4 : activityReferencesCreated
5 : createManifestFile()
6 : getActivities()
7 : getActivities()
8 : returnListOf Activities
9 : getTransition()()
10 : returnTransition
11 : manifestfileCreated

Figure 8: UML sequence diagram: source view of the LAMS package creation

UML sequence diagram that represents how LAMS package was built in source is shown in Fig. 8. First, "*createLAMSPackage*" function creates new folder, and after that calls "*createActivityReferences*" function which creates a folder with *tool.xml* file for each LAMS activity. Because Noticeboard activity's *tool.xml* file needs to be updated, this function calls the "*creteToolXMLForNoticeboard*" function for each Noticeboard activity. After the "*createActivityReferences*" function finishes its work and goes back to the "*createLAMSPackage*" function, all tool.xml files are created and prepared to be referenced from manifest file. Now, the "*createLAMSManifest*" function is called. First, this function creates a new manifest file named *learning_design.xml* and its header. The header contains some global information about LAMS sequence, such as the first activity in sequence, creation date, etc. After that the "*createLAMSManifest*" function calls the "*getActivities*" function. This function generates activities part in manifest file. After activities part is created, the "*getTransition*" function is called for each connected pair of activities. After the manifest file is created, the "*createLAMSPackage*" function compresses created folder to zip file. That zip file is actually a LAMS package and it may be imported to any LAMS installation.

5. FURTHER IMPROVEMENTS

As mentioned above, at the moment it is possible to create only default LAMS activities except Noticeboard. The next improvement of QDITA application would be to create editors for each LAMS activity and set its values directly from QDITA.

Another improvement would be to establish communication between QDITA and LAMS installation, and to import created LAMS package into LAMS directly from QDITA.

6. CONCLUSION

In order to create the best possible e-learning courses, well structured with all required activities, it was needed to include two different systems, QDITA and LAMS, and to integrate them in one unit. Issues of QDITA and LAMS integration were detected and one possible solution was found. This solution was created, tested and explained in this paper.

LITERATURE

[1] OASIS DITA Standard Version 1.2 - http://docs.oasis-open.org/dita/v1.2/spec/DITA1.2-spec.html

[2] Veljković D., Nikolić B., Dinić M., QDITA: An Eco-System For Adaptive E-Learning, The Third International Conference on e-Learning (eLearning-2012) 27-28 September 2012, Belgrade, Serbia

[3] IMS Learning Design http://www.imsglobal.org/ learningdesign/

[4] LAMS - http://lamsfoundation.org

[5] DITA specialization, http://docs.oasis-open.org/dita/v1.0/ archspec/ditaspecialization.html

[6] DITA Open Toolkit (OT) - http://www.ditaopentoolkit.org