A Method on Single Source Publishing for Music in DITA

Xuhong Liu, Yunmei Shi, Peng Liu, Ning Li
Computer School, Beijing Information Science & Technology University,
Beijing, China
E-mail: liuxh0315@sina.com.cn

***Abstract*—Federated Media Publishing means to distribute the single source content through multimedia after integrating various media manners, which is the trend of digital publishing. DITA is designed to produce multiple deliverable formats from a single set of DITA content, but it can't be directly used in Federated Media Publishing for not coping well with multimedia except for text and image. This paper takes music as an example to discuss the method of integrating multimedia into DITA. The specialization facility is used to integrate MusicXML DTDs into DITA and a solution is putted forward to provide appropriate rendition distinctions according to audio and five-line staff. Experiments show the solution is efficient to extend DITA for dealing with multimedia, which can be used in federated media publishing.**

Keywords—Darwin Information Typing Architecture; Federated Media Publishing; MusicXML; Single Source

I. INTRODUCTION

During the last ten years, Federated media publishing industry in china has been developed rapidly with the mobile terminal widely used and the popularity of the Internet. Federated media publishing industry is a new form and growing tendency of digital publishing. Federated media publishing means to distribute the single source content through multimedia after integrating various media manners [1]. Based on a single source file contains music information, for example, appropriate rendition distinctions are provided according to audio and five-line staff. The federated media publishing needs to establish perfect content creation and management mechanism to manage all kinds of digital contents for single source publishing.

There exists several techniques used for content organizing, which can be divided into three types, paginated document, flow document and the combination of paginated and flow document. The typical paginated documents include PDF(Portable Document Format), XPS((XML Paper Specification) and CEB(Chinese eBook), the typical flow documents include TXT, HTML(Hypertext Markup Language), DOC\DOCX (Microsoft Office Open XML) and Epub(Electronic Publication) as so on.. The combination of the above two types is CEBX(Common e-Document of Blending XML) as so on. However, the content organizing forms above are not suitable for federated media contents, as the content can't be disassembled and reused after the whole content is established, which results in content redundancy.

With the development of digital publishing, other forms of content organizing appeared, such as DITA（Darwin Information Typing Architecture）which is developed by IBM. DITA is an XML-based architecture for authoring, producing and delivering topic-oriented, information-typed content that can be reused and single-sourced in a variety ways [2-3]. IBM donated DITA to the OASIS standards organization in 2004. DITA is widely used by technical enterprise currently in technology documents authoring, digital publishing and enterprise informationization (for example knowledge management, content management and document management) as so on [4].

Although DITA provides a better solution for content management and reuse, it can't be directly used for federated media publishing as not supporting multimedia very well. Therefore, it is necessary to extend DITA architecture to support multimedia and be used to federated media publishing in the future.

The publication of music is very complex for rich diversity of presentation forms, therefore, this paper take music as an example to discuss the technology to extent DITA architecture for supporting multimedia, and then can be used in federated media publishing in the future. So far as is known to the writer, there is no related research in this subject yet

DITA-aware commercial tools include Arbortext by PTC, FrameMake by Adobe and so on. The DITA Open Toolkit is an implementation of the OASIS DITA Technical Committee's specification for DITA DTDs and Schemas. The Toolkit transforms DITA content (maps and topics) into deliverable formats [5].

MusicXML is an XML-based file format for representing western musical notation. MusicXML is developed by Recordare LLC, It is designed for the interchange of scores, particularly between different score writers [6].

This paper discusses the method to integrate musicXML into DITA OT after musicXML Specialization, and of implementation single source publishing for DITA document that contains musicXML content. The DITA document can be delivered in different target media according to user's requirement.

II. DITA OVERVIEW

A. DITA Application Architecture

DITA is sophisticated XML-based application architecture for authoring, producing, and delivering information. DITA has predefined DTDs and Schema document shells necessary for digital publishing based on which the workflow from content authoring to content delivery is established. The whole process is illustrated in figure 1, specific processes are as follows

(1) Specialization (optional). The specialization feature of DITA allows for the creation of new element types and

Published by Atlantis Press, Paris, France.

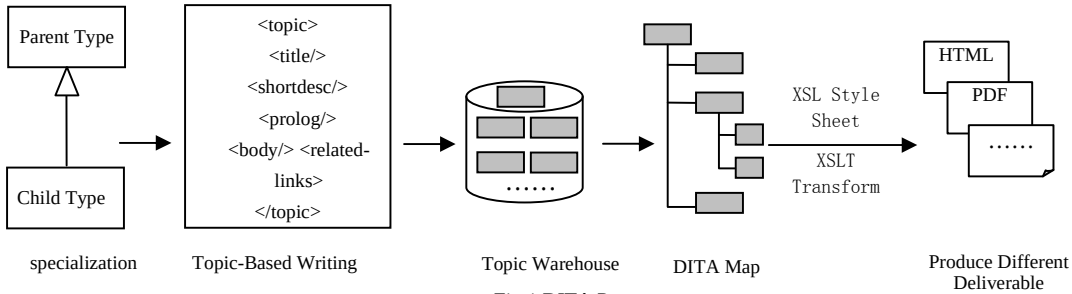

Fig.1 DITA Process

attributes that are explicitly and formally derived from existing types. The resulting specialization allows for the blind interchange of all conforming DITA content and a minimum level of common processing for all DITA content. It also allows specialization-aware processors to add specialization-specific processing to existing base processing.

(2) Topic-Based Writing. Separate information into appreciate granularity content chunk to generate different topic types. A topic has just enough content to make sense by itself but not so much content that it covers more than one procedure, once concept, or one type of reference information.

(3) Topic Warehouse. The contents are stored according to their topic types; each topic is stored as a single unit which composes the topic warehouse.

(4) DITA Map. The topics chosen from topic warehouse are assembled to DITA document according to DITA map.

(5) Production of Different Deliverable. DITA is designed to produce multiple deliverable formats from a single set of DITA content. DITA supports the separation of content from presentation. The style sheet and XLST transformations produce different deliverable formats from a single source.

B. DITA Specialization

In DITA, a topic is the basic unit of authoring and reuse. A topic is a unit of information that describes a single task, concept, or reference item. The information category (concept, task, or reference) is its information type (or infotype). A new information type can be introduced by specialization from the structures in the base topic DTD. Typed topics are easily managed within content management systems as reusable, stand-alone units of information.

DITA provides two kinds of specialization, one is DITA typed topic specializations (infotyped topics), the other is DITA vocabulary specialization (domains) [7].

The typed topics represent the fundamental structuring layer for DITA topic-oriented content. The basis of the architecture is the topic structure, from which the concept, task, and reference structures are specialized. Extensibility to other typed topics is possible through further specialization. The four information types (topic, concept, task, and reference) represent the primary content categories used in the technical documentation community. Figure 2 illustrates the relationship between the four topics. Moreover, specialized information types, based on the original four, can be defined as required.

Using the same technique as specialization for topics, DITA allows the definition of domains of special vocabulary that can be shared among infotyped topics. Domain specialization doesn't define new typed topics but new elements and attributes in vocabulary module..

The two specializations both define elements and entities in module files(.mod and .ent file), then integrate them into DTD files of special topic type, the specialization document shell includes new vocabulary module and all ancestor's module files, so the elements defined by ancestor are valid in the new DTDs. Moreover, special code module and XSL style sheet for processing new elements and attributes should be provided in deliverable if necessary.

C. DITA One-Source Publishing

During the fifth step in DITA process, DITA OT provides a series XSL style sheets to produce different deliverable forms from a single source. Figure 3 provides an overview of the processing and publishing of DITA documents using DITA Open Toolkit:

(1) In step 1, the Ant build tool is initialized (either through an optional batch script or directly from the command line), and arguments that will be passed from the Ant script to Ant are validated.

(2) In step 2, the Ant build tool calls the Toolkit, which produces the rendered temporary files. Input to this step is the .dita and .ditamap source files, and the DITA project DTDs or schemas.

(3) In step 3, the XSLT processor (SAXON or Xalan) produces the published output files. Input to this step are the XSLT style sheets for the project, and the temporary files produced in step 2.

There are two reasons for DITA architecture not supporting music very well according to above process. Firstly, DITA can't recognize elements defined in musicXML. Secondly, DITA doesn't provide code module and style sheet for processing musicXML elements. So the two problems must be resolved in order to extend DITA to support music. The following in this paper discusses the resolution in detail.

III. MUSICXML SPECIALIZATION

The essence of DITA architecture is inheritance which enables reuse the existing information and code module in DITA architecture. User need only concentrate on the new DITA elements. New DITA element should inherit from existing DITA elements and so can be mapped to its ancestor. Every element type exists in a specialization hierarchy, which goes from the base module (topic or map) through any intermediate modules to the element itself. Every DITA element must have a @class attribute. The value of the class attribute is the specification of the specialization hierarchy for the element. The new element can use the translation rules for its parents.

Elements related to music notation are defined in musicXML, but they have no class attributes identify their specialization hierarchy which means the elements in musicXML are non-DITA elements. Therefore, the <foreign> element should be specialized to integrate the non-DITA elements in musicXML. Specialization of the <foreign> element is an open extension to DITA for the purpose of incorporating standard vocabularies for non-textual content. The <foreign> element can be specialized both in topic specializations and domain specializations [8]. This paper chooses domain specialization as an example to illustrate the process.

(1) Establish Music Domain Module Files

① Define music domain module's name, such as "musicxml-d", which is related to the specialization hierarchy identified by class attribute of the new element.

② Integrate MusicXML DTDs into music domain module and establish the new element <musicxml> as listed in the following code. <score-partwise> is the child element of <musicxml> while the root element in musicXML. The non-DITA elements defined in MusicXML DTD are integrated by this way.

③ Define class attribute of the new <musicxml> element, this attribute shows the <musicxml> element belongs to musicxml-d and inherited from <foreign> element in topic. It is worth noting that only <musicxml> is DITA element, the other elements in musicXML DTD, such as <score-partwise>, are non-DITA elements, so have no class attribute.

The main code in music domain module file is listed as follows.

```
<!—main code in music domain module file -->
<!-- integrate MusicXML DTD -->
<!ENTITY % musicxml-partwise_dtd SYSTEM "partwise.dtd" > %musicxml-partwise_dtd;
<!—define new element for music domain-->
<!ENTITY % musicxml.content "((score-partwise)?)">
<!ENTITY % musicxml.attributes ' '>
<!ELEMENT musicxml %musicxml.content;>
<!ATTLIST musicxml %musicxml.attributes;>
<!—declare new element's class attribute -->
<!ATTLIST musicxml %global-atts; class CDATA "+ topic/foreign musicxml-d/musicxml">
```

(2) The music domain module in (1) can be shared by any DITA topic type, see reference [9] for details。

IV. IMPROVE DITA CONTENT DELIVERY

User can author DITA document that contains music information after above specialization. The following DITA document is concept topic type integrating music domain module, <conbody> which contains the main body of the document is defined in concept topic type. Music "happy new year" is nested into the document by <musicxml> element, only part of the document is listed due to limited space.

```
<!—concept topic type DITA document integrated with music domain module -->
<!DOCTYPE concept SYSTEM "dtd\concept.dtd">
<concept id="testmusicdomain">
 <conbody>
  <musicxml>
      ... ...
   <score-partwise>
    <part-list>
     <score-part id="P1">
      <part-name>Piano</part-name>
      <part-abbreviation>Pno.</part-abbreviation>
      <score-instrument id="P1-I3">
       <instrument-name>Piano</instrument-name>
      </score-instrument>
     </score-part>
    </part-list>
    <part id="P1">
    <measure number="1" width="191.55">
     <note-music default-x="83.43" default-y="-35.00">
      <pitch>
       <step>F</step>
       <octave>4</octave>
      </pitch>
      <duration>1</duration>
      <voice>1</voice>
      <type>eighth</type>
      <stem>up</stem>
      <beam number="1">begin</beam>
     </note-music>
     ... ...
    </measure>
        <measure number="2" width="120.12">
     <note-music default-x="12.00" default-y="-25.00">
      <pitch>
       <step>A</step>
       <octave>4</octave>
      </pitch>
      <duration>1</duration>
      <voice>1</voice>
      <type>eighth</type>
      <stem>up</stem>
      <beam number="1">begin</beam>
     </note-music>
        </measure>
    ... ...
  </score-partwise>
  </musicxml>
 </conbody>
</concept>
```

As noted before, the new DITA element and attribute must be established based on those in the more general topic type, and then the transformation rules for ancestor element and attribute can also be used to the new DITA element and attribute. Therefore, one can process <musicxml> element in the same way as <foreign> element. However, DITA doesn't

provide support for music and can't translate <musicxml> element and its descendants accurately. Therefore, the new special translation rules and code module should be added to DITA for processing the specialization element.

Music can be presented as audio and five-line staff according to user's requirements; the paper will discuss the method to generate two presentation forms.

A. *Presentation in Vision Media*

If the target document is PDF and HTML, the music information in DITA document should be presented as five-line staff nested in the target.

Many document formats and tools, such as PDF reader and explorer, don't support musicXML currently, but may recognize SVG markup language or image. Two different solutions are designed according to whether or not the target document supports SVG.

For those target documents formats that support SVG, like HTML5, can translate the <musicxml> content in DITA document to SVG, and then embedded within target document, the details are as follows.

(1) Extract <musicxml> element and its all descendant element; build xsl translation according to reference [10] for translating the selected elements into SVG format.

(2) Replace <musicxml> content in DITA document with SVG format content obtained in the last step.

(3) Translate DITA document by existing xsl style sheet in DITA OT to different formats according to user requirement.

If the target document is PDF, one can translate <musicxml> content to FO file including SVG by xsl translation file designed in the first step, as FOP provides function to translate FO document that contains SVG to PDF.

The xsl file built here need to be deployed to DITA OT as a plug-in, see reference [5] for details.

For those target documents formats that not support SVG, as almost all document formats support image, the <musicxml> content can be translated to five-line staff image and then embedded in target document by the following procedure.

(1) Extract <musicxml> element and its all descendant element and then translate them to SVG.

(2) Rasterize SVG by Apache Batik and Rasterizer Ant task. The two tools are open and can translate SVG file to four formats—PNG, JPEG, TIFF, PDF[11].

(3) Replace the <musicxml> content in DITA document with rasterizing five-line staff image.

(4) Publish DITA document by existing xsl file in DITA OT.

B. *Presentation in Audio Media*

If target format is audio, solution for deliver DITA document that contains music information to audio file should be built.

There are several audio formats among which MIDI (Musical Instrument Digital Interface) is an electronic musical instrument industry specification that enables a wide variety of digital musical instruments, computers and other related devices to connect and communicate with one another. It is a set of standard commands that allows electronic musical instruments, performance controllers, computers and related devices to communicate, as well as a hardware standard that guarantees compatibility between them. MIDI file is very compact and can be translated to other format, so the following will discuss the solution of translation DITA document including music information to MIDI file.

The key problem to be resolved is to write code module for translating DITA document including music information to audio format, namely, a java class for outputting audio format file should be built and deployed to DITA OT. The java class must extend org.apache.tools.ant.Task, and implement translation function in method execute(), because DITA OT processes documentation project as an Ant project. The details are as follows:

(1) Extract <musicxml> element and its all descendant element.

(2) Take the content of each <measure> element in <musicxml> as a measure in music score. Take the content of each <note-music> element as musical note in music score. Then, build MIDI message that will be integrated into track according to measure and note information, the output MIDI file is generated finally.

The new java class will be treated as a new rule for processing <musicxml> element and be deployed to DITA OT by the extension points for plug-in mentioned in reference [5]. DITA OT can deliver audio file now.

V. IMPLEMENTATION AND TEST

A single source DITA document including music information can be delivered to different formats, such as HTML, PDF and audio format, by extending DITA OT to support musicXML.

Figure 4 illustrates the presentation of music information in HTML file which is transformed from the DITA document listed in section 3, user can browse five-line staff and play audio in browser. Figure 5 is the presentation result of music information in PDF document that transformed from the same DITA document.

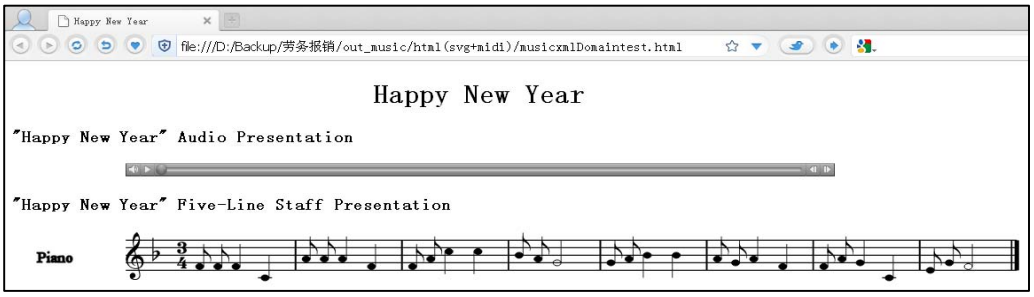

Fig.4 Presentation of Music Information in HTML File

Fig.5 Presentation of Music Information in PDF File

The experiment results show that the method provided in this paper can extend DITA to support music information and produce different deliverable forms from a single source including music score. This provides insights to integrate other multimedia into DITA.

VI. CONCLUSIONS AND FURTHER RESEARCH

The federated media publishing needs to establish perfect content creation and management mechanism to manage all kinds of digital contents for single source publishing. Although DITA provides a better solution for content management and reuse, it can't be directly used in federated media publishing for not supporting multimedia very well. This paper puts forward a method to extend the DITA architecture to support music score; the single source DITA document can produce different deliver forms according to user's requirement. The solution in this paper can be extended to other multimedia.

Furthermore, the solution in this paper does not take account of synchronism problem in multimedia, karaoke, for example, how to translate DITA document including music score, video and lyrics, to different deliverable? How to represent them synchronously? The future research will resolve these issues.

ACKNOWLEDGMENT

This work is supported by funded project in Funding Project for Academic Human Resources Development in Institutions of Higher Learning Under the Jurisdiction of Beijing Municipality under the Grant PHR201008439 from Beijing Municipal Education Commission, China.

This work is also supported by funded project in Research on key technology of multimodal output for music in DITA under the Grant KM201311232012 from Beijing Municipal Education Commission, China. We would like to thank the funding agency in supporting this research.

We would like to thank the funding agency in supporting this research.

REFERENCES

[1] Y. Zhang, "Federated media publishing: Present Status and Future Development," Modern Publishing. Beijing, vol. 2, pp. 14-17, 2011.

[2] OASIS, "OASIS Darwin Information Typing Architecture (DITA)" TC. (2009-5-12). http://www.oasis- open. org/ committees/dita/

[3] M. Priestley, "DITA XML: a reuse by reference architecture for technical documentation," Proceedings of the 19th annual international conference on Computer documentation, ACM Press, 2001, pp. 152 – 156.

[4] F. Wei, "A Study on Darwin Information Typing Architecture," Journal of Intelligence, Xi'an, vol.28 pp. 172-175, 2009.

[5] DITA Open Tool. (2012-9-17). http:// dita-ot.sourceforge.net/

[6] L.M. Surhone, M.T. Tennoe, S.F. Henssonow. Musicxml. Saarbrucken, Germany: VDM Publishing House Ltd, 2010

[7] M. Priestley, D.A. Schell, "Specialization in DITA: Technology," Process, & Policy, ACM International Conference on Design of Communication, ACM Press, 2002, pp. 57-64.

[8] Foreign generalization. (2007-8-1). http:// docs.oasis-open.org/dita/v1.1/OS/archspec/foreigngeneralization.html

[9] K. Eliot, "DITA for Practitioners Volume 1: Architecture and Technology," USA: XML Press, 2012.

[10] MusicXML to SVG. (2011-11-2). http:// zh.sourceforge.jp/projects/sfne_music xmltosvg/

[11] Batik SVG Toolkit. (2010-2-2). http:// xmlgraphics.apache.org/batik/