UNIVERSITY
OF WOLLONGONG
AUSTRALIA

University of Wollongong
Research Online

Faculty of Informatics - Papers (Archive)

Faculty of Engineering and Information Sciences

2006

On-demand partial schema delivery for multimedia metadata

S. J. Davis
University of Wollongong, stdavis@uow.edu.au

I. S. Burnett
University of Wollongong, ianb@uow.edu.au

Publication Details

This paper was originally published as: Davis, SJ & Burnett, IS, On-demand partial schema delivery for multimedia metadata, IEEE International Conference on Multimedia and Expo 2006, Toronto, Canada, July 2006, 1513-1516. Copyright 2006 IEEE.

Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: research-pubs@uow.edu.au

On-demand partial schema delivery for multimedia metadata

Abstract
XML is a popular approach to interoperable exchange of Multimedia metadata between a wide range of devices. This paper explores extending the use of the Remote XML Exchange Protocol (previously proposed by the authors) as a mechanism to provide efficient interaction with complex Multimedia XML documents and their associated schemas. This is particularly applicable to users with limited application complexity devices and/or limited bandwidth connections. Many XML documents do not fully utilize all the information present in a given schema; thus, users download substantial redundant information for the current application. This paper introduces the use of RXEP for the transmission of small, relevant schema sections. The paper investigates the advantages of schema retrieval using RXEP in terms of the bandwidth saved.

Disciplines
Physical Sciences and Mathematics

Publication Details
This paper was originally published as: Davis, SJ & Burnett, IS, On-demand partial schema delivery for multimedia metadata, IEEE International Conference on Multimedia and Expo 2006, Toronto, Canada, July 2006, 1513-1516. Copyright 2006 IEEE.

This conference paper is available at Research Online: http://ro.uow.edu.au/infopapers/486

ON-DEMAND PARTIAL SCHEMA DELIVERY FOR MULTIMEDIA METADATA

Stephen J. Davis[1] *and Ian S. Burnett*

School of Electrical, Computer and Telecommunications Engineering
University Of Wollongong, Australia

ABSTRACT

XML is a popular approach to interoperable exchange of Multimedia metadata between a wide range of devices. This paper explores extending the use of the Remote XML Exchange Protocol (previously proposed by the authors) as a mechanism to provide efficient interaction with complex Multimedia XML documents and their associated schemas. This is particularly applicable to users with limited application complexity devices and/or limited bandwidth connections. Many XML documents do not fully utilize all the information present in a given schema; thus, users download substantial redundant information for the current application. This paper introduces the use of RXEP for the transmission of small, relevant schema sections. The paper investigates the advantages of schema retrieval using RXEP in terms of the bandwidth saved.

1. INTRODUCTION

XML [1] brings a wealth of advantages to application communication, but it is often said to be verbose. While this is not of vital importance in all applications, in the mobile environment and on devices with limited processing power or memory, the verboseness can be a significant impediment. The structure of XML documents is often expressed using an XML expressed schema (XML schema [2]) which allows applications to check and validate instances of documents. However, such schemas also bring with them the problems of verbosity; a single XML document can often link to several, potentially large, schemas. While it is not always necessary to possess the schema to receive an XML document, valid modification of the XML requires the schemas to be available. This paper proposes the use of a simple protocol (RXEP [3]) which allows the retrieval of small subsections of XML documents. The key to RXEP is to allow users to retain the advantages of the structure of XML and XML schema while minimizing the overhead in exchanging the raw data (which, after all, is what the user is really wanting to access). Extension of RXEP makes it applicable to schema retrieval and provides a mechanism for full usage of XML documents and schemas in e.g. mobile environments.

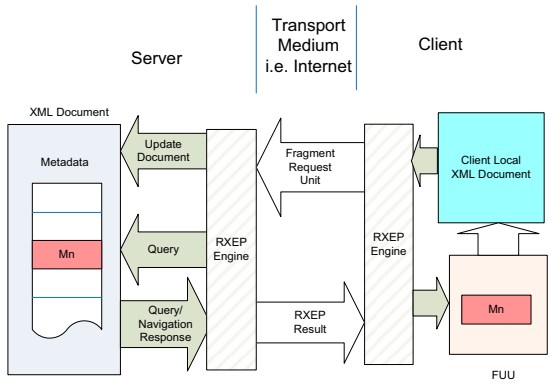

Figure 1 - Interaction of FRUs and RXEP FUUs [7]

While the RXEP protocol can be used generally for XML, this paper focuses on its usage within the exchange of multimedia descriptions. Many descriptors are available for the description of multimedia content, and consequently, many XML schemas have been generated. For example, schemas exist representing albums, pictures, music, audio descriptors and movies etc. Some descriptors have been standardized such as those descriptors found in standards such as MPEG-7 [4] and MPEG-21 [5]. Furthermore, some users find these standardized descriptors inadequate and prefer to create their own descriptors or extend current sets, and have thus generated user defined schemas.

Increasingly, communities of users are sharing and accessing content on the Internet. Alongside the content itself, there is increasing usage of accompanying metadata which allows the community to label the content according to shared schemas. Whether the community chooses standardized descriptors (i.e. MPEG-7) or their own creations, a large number of XML documents and schema documents must be exchanged between the community's devices. These large, redundant descriptor sets are a significant problem in mobile environments where data is usually charged per kilobyte. For communities to operate effectively, all users need to be able to contribute to the growth of data, and hence editing of the descriptors is an

[1] Partially funded by the Smart Internet Technology CRC

important part of the process. However, editing of XML which is conformant to a schema requires access to the schema, and hence the transmission and storage of a large XML document.

The size of schemas was recognized as a problem by the MPEG community while standardising MPEG-7 with the result that MPEG-7 TEM and BiM [4] offer standardized techniques for unidirectional transmission of schema fragments. However, this mechanism is of little use in a collaborative environment where users must have the ability to access fragments of schema as XML documents are parsed and processed.

To allow mobile device users to exploit XML without these penalties, this paper employs an extended RXEP protocol to allow fragments of schemas to be delivered with requested fragments of XML. Utilizing this schema request technique, users may create rich and vast sets of multimedia data descriptors with the knowledge that their schemas can be used in both high and low bandwidth environments.

2. REMOTE XML EXCHANGE PROTOCOL

The Remote XML Exchange Protocol (RXEP) [3] is designed to handle the underlying delivery of XML Fragments. An illustration of RXEP is shown in Figure 1.

Clients which implement RXEP requests are able to query (i.e. xpath [6]) and browse (navigate) through remote XML documents, retrieving only relevant document fragments. This introduces significant savings as it avoids the user retrieving the entire XML document even if only a small section of that document is desired. RXEP commands are defined using XML schema, and is broken down into two parts: upstream commands (RXEP Fragment Request Units) [3,7] and downstream commands (RXEP Fragment Update Units [7]).

2.1 Fragment Request Units

Fragment Request Units (FRUs) are created by the users to request fragments of XML from a remote XML document. The FRUs are created in XML (valid to the FRU schema) from a selection of RXEP commands. Briefly, basic FRU commands are as follows: Src, Query, XMLPull and Stream. These commands allow a client to select a document, query a document, issue XML Pull commands on a document and stream sub-branches of a document (for further details see [7]).

FRUs are capable of requesting any fragment of the XML document (based on fragment size and location), thus providing clients with random access. This allows a client to jump into any node in an XML document, or to simply, “navigate backwards” (such an operation may be entirely client side if previous XML fragments have been cached locally). A sample RXEP FRU requesting the child nodes of the */Media/Music/Song[1]* node (see Figure 2) is shown in Figure 3.

```
<Media xmlns="mediaNS:2004">
    <Music>
        <Song id="Hit1">
            <Title>Hit.1</Title>
            <Description>Song 1</Description>
            <Artist>A. Artist</Artist>
            <Format>MP3</Format>
            <Length>02:23</Length>
        </Song>
        <Song id="Hit2">
            <Title>Hit.2</Title>
            <Description>Song 2</Description>
            <Artist>B. Artist</Artist>
            <Format>OGG</Format>
            <Length>03:46</Length>
        </Song>
    </Music>
</Media>
```

Figure 2 - Example XML Document

2.2 RXEP Fragment Update Units

RXEP Fragment Update Units (RFUUs) are XML containers (defined by an XML schema) of commands instructing the client to update parts of an XML Document. RFUUs define commands such as: Add, Delete, Update and Insert (for further details see [7,8]).

An example of an RXEP FUU, as a response to the RXEP FRU in Figure 3, is illustrated in Figure 4. This RFUU demonstrates addition of the XML under the *ADD* node to the location specified by the XPath locator, in this case */Media/Music/Song[1].*

3. SCHEMA FRAGMENTATION

XML schemas may contain information that is not required by receiving user. Through fragmentation of an XML schema, only relevant portions of the schema(s) can be delivered. Users could simply request portions of the schema as needed. Intelligent servers could extend this concept further and make decisions based on the current usage of the XML document and deliver a relevant schema fragment(s) along with the corresponding XML fragment(s).

Since an XML schema document is a well formed XML document, RXEP becomes an ideal candidate for the basis of a schema fragmentation and delivery mechanism. An example portion of a schema used to validate the XML in Figure 2, is shown in Figure 5.

Although RXEP could be directly used on XML schema documents, this would require many separate RXEP requests to be made for each schema. Thus, we are more interested in simultaneously building the client side set of

```
<FRU>
        <XPath location="/Media/Music/Song[1]">
</FRU>
```

Figure 3 - Example RXEP FRU

```
<RXEP>
    <ADD location="/Media/Music /Song[1]">
        <Title>Hit.1</Title>
        <Description>Theme song</Description>
        <Artist>A. Artist</Artist>
        <Format>MP3</Format>
        <Length>02:23</Length>
    </ADD>
<RXEP>
```

Figure 4 - Example RXEP FUU result from the FRU as in Figure 3

schemas when navigation or queries are performed on a remote XML document. Furthermore, when an XML fragment is requested, the server or peer can determine what corresponding schema fragments are needed and can be delivered to the client alongside the XML fragment.

Dynamic creation of client-side schemas require fragmentation commands such as add, insert, update and delete, which is already provided by RXEP. In order to identify the correct schema to perform these operations, the RXEP instruction schema is extended to allow an additional attribute: SchemaNS – which defines the schema Namespace to modify.

With this simple extension, RXEP now provides a client with three options for the reception of both XML and schema fragments:

a) Client Requested **-** This allows a client to request a fragment (and in particular, schema fragments) only when necessary. For example, a user listening to their favorite music album may wish to modify a song title descriptor. Using RXEP, the user can request the XML schema fragment for the nodes that require changes so as to generate a valid, edited descriptor.

b) Server Determined - Necessary fragments are determined by the server and delivered with the requested XML fragment within a single RXEP unit. For example, when a fragment of XML is requested, the relevant fragment of schema XML is also packaged inside the RXEP container. This concept is further illustrated in Figure 6 where Peer A is holding the XML and corresponding schemas which Peer B is requesting. As illustrated, RXEP delivers small XML fragments and schema fragments, hence building a partial document and schema locally at Peer B.

Additionally, intelligent servers may pre-empt usage (either based on current usage, or statistics of previous clients), and deliver additional fragments of XML and/or schema upfront.

c) Combination - Some clients may wish to cache locally schema fragments previously received. When a sub-tree of XML in which the corresponding schema fragments are not yet retrieved, the client can notify the server to "push" schema fragments for a desired sub-tree (as defined in the Server Determined case)

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="mediaNS:2004"
xmlns:ns1="mediaNS:2004">
 <xs:element name="Media">
  <xs:complexType>
   <xs:sequence>
    <xs:element ref="ns1:Music"/>
    <xs:element ref="ns1:Videos"/>
   </xs:sequence>
  </xs:complexType>
 </xs:element>
 <xs:element name="Music">
  <xs:complexType>
   <xs:sequence>
    <xs:element maxOccurs="unbounded"
          ref="ns1:Song"/>
          …..
   </xs:sequence>
  </xs:complexType>
 </xs:element>
 <xs:element name="Song">
  <xs:complexType>
   <xs:sequence>
    <xs:element name="Title" type="xs:string"/>
    <xs:element name="Description" type="xs:string />
    <xs:element name="Artist" type="xs:string />
    <xs:element name="Format" type="xs:string />
    <xs:element name="Length" type="xs:string />
   </xs:sequence>
   <xs:attribute name="id" use="required"
          type="xs:NCName"/>
  </xs:complexType>
 </xs:element>
</xs:schema>
```

Figure 5 - Example XML Schema for a Simple Media Album

3.1. Schema XPath Locators

The location attribute plays a vital role in RXEP FUU commands. This attribute defines the "location" in the local version of the schema at which the incoming FUU modification is to be made. The location attribute uses XPath notation, and we refer to these as XPath Locators. Whilst XPath locators can be created for an XML schema instance, they can prove to be lengthy expressions. As an example and referring to Figure 5, if the *Title* element is to be referenced, the standard XPath locator would be: */xs:Schema/xs:Element[@name="Song"]/xs:complexType/ xs:Sequence/xs:Element[@name="title"].* Fortunately, some assumptions can be made about the schema XML to reduce the complexity of the RXEP XPath expressions. Utilizing some of the rules defined in XML schema and considering RXEP query and navigation commands in particular, the following assumptions and simplifications can be applied:

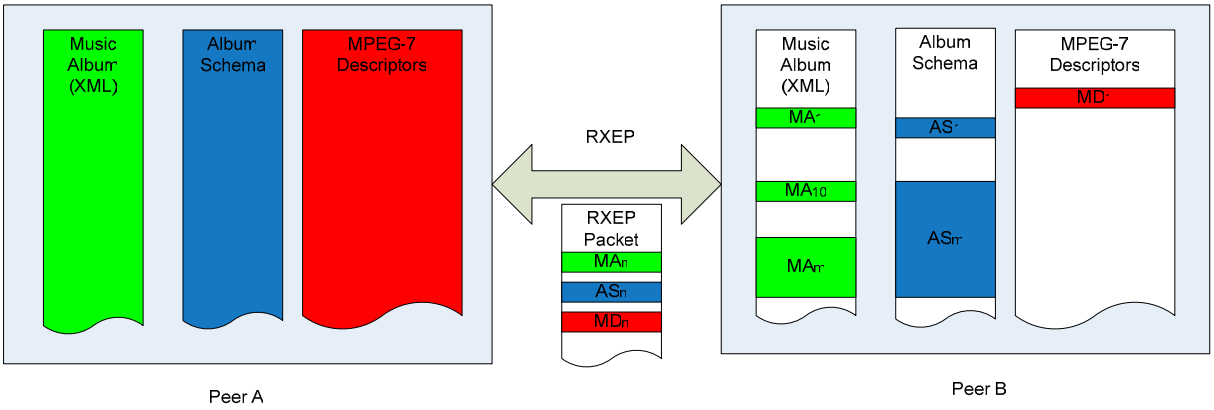

Figure 6 – Illustration of delivering an XML fragment and Schema fragments in one RXEP Packet

- When requesting XML fragments, only elements are selected, and thus, the locators can be restricted to specify only elements (i.e. XML schema attributes cannot be selected);
- Following from the above, there is no need to specify <element name="name">, but rather specify just the defined name (e.g. /name/);
- A choice/sequence/all node cannot contain multiple child elements with the same element name [2];
- Modelgroup [2] nodes (i.e. choice, sequence and all) must be declared inside a complexType tag and thus the complexType portion can also be omitted;
- The first schema element (i.e. *<xs:Schema>*) can be omitted as this must be present to represent a valid schema; and

Whilst observing these rules when using RXEP to partially receive XML schemas, the schema XPath locator from our previous example can be reduced to: */Song/[Sequence]/Title*.

4. EXAMPLE SCENARIOS

To demonstrate the effectiveness of RXEP when applied to both XML documents and their corresponding XML schemas, the following discusses some practical examples.

The first scenario investigates the Server determined delivery of navigated XML and its XML schema. This scenario refers to the XML in Figure 2 and its corresponding schema in Figure 5. The first RXEP request to the XML delivers the *<Media>* tag and the corresponding schema fragment. The user chooses the Music tag, and then the Song tag. On the client side, the User has a partial schema containing only the Music related portions. The received file would look similar to Figure 5, noting, that in this case the "…" represents schema fragments not retrieved.

Another scenario considers a user browsing though an XML document on their mobile device in a low bandwidth environment. Upon browsing, the user notices that a Video clip is missing some important 'performer' metadata and wishes to make the addition. To ensure that the metadata is conformant to the schema, the user may issue an RXEP request to retrieve the small fragment of 'performer' metadata schema to ensure that the newly entered XML remains valid and conformant. The user thus avoids downloading the entire schema when they only need access to a small portion.

6. CONCLUSION

This paper has demonstrated the application of RXEP to XML schemas as a means to transmit only the required parts in order to validate partially received XML Multimedia documents. Utilizing this technique, a peer can deliver both a requested fragment and its corresponding schema within a single RXEP packet. This is important as it allows portable device, users to exploit Multimedia metadata without the penalty of downloading and/or storing large XML schema documents.

7. REFERENCES

[1] W3C, "Extensible Markup Language (XML)," [online document] http://www.w3.org/XML/.

[2] W3C, [online document] "XML Schema," http://www.w3.org/XML/Schema

[3] S. Davis and I. Burnett, "Efficient Delivery in the MPEG-21 Framework", *1st International Conference on Automated Production of Cross Media Content for Multi-channel Distribution (AXMEDIS)*, Italy, 2005

[4] Information technology, "Multimedia content description interface -- Part 1: Systems," ISO/IEC 15938-1:2002.

[5] Bormans and K. Hill, "MPEG-21 Overview v.5," ISO/IEC JTC1/SC29/WG11/N5231, October 2002.

[6] W3C, "XML Path Language (XPath)," [online document] http://www.w3.org/TR/xpath

[7] S.J. Davis and I.S. Burnett, "Collaborative Editing using an XML Protocol," *IEEE TENCON'05*, Melbourne, 21 - 24 November, 2005

[8] S.J. Davis and I.S. Burnett, "Exchanging XML Multimedia Containers Using a Binary XML Protocol," *IEEE International Conference on Multimedia & Expo (ICME) 2005*, Amsterdam, July 6-8, 2005.