Novel Approaches to the Delivery of XML and Schemas

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

Follow this and additional works at: https://ro.uow.edu.au/theses

Recommended Citation

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
NOTE

This online version of the thesis may have different page formatting and pagination from the paper copy held in the University of Wollongong Library.

UNIVERSITY OF WOLLONGONG

COPYRIGHT WARNING

You may print or download ONE copy of this document for the purpose of your own research or study. The University does not authorise you to copy, communicate or otherwise make available electronically to any other person any copyright material contained on this site. You are reminded of the following:

Copyright owners are entitled to take legal action against persons who infringe their copyright. A reproduction of material that is protected by copyright may be a copyright infringement. A court may impose penalties and award damages in relation to offences and infringements relating to copyright material. Higher penalties may apply, and higher damages may be awarded, for offences and infringements involving the conversion of material into digital or electronic form.
Novel Approaches to the Delivery of XML and Schemas

A thesis submitted in fulfilment of the requirements for the award of the degree

Doctor of Philosophy

from

UNIVERSITY OF WOLLONGONG

by

Stephen James Davis
Bachelor of Engineering (Honours Class I)
University of Wollongong, 2001

SCHOOL OF ELECTRICAL, COMPUTER
AND TELECOMMUNICATIONS ENGINEERING
2007
Abstract

Typically XML documents are delivered as whole documents, and the transmission does not consider if all of this data may actually be relevant to the user. This results in inefficiencies in terms of both bandwidth (transferring unnecessary data) and computing resources (extra memory and processing to handle the entire XML document). Through exploitation of XML's tree-like structure, a simple and lightweight protocol is introduced (referred to as RXPP). Designed with mobile devices in mind, RXPP provides users with the ability to navigate and retrieve data from remote documents on a node-by-node or branch-by-branch basis, allowing users to retrieve only fragments of interest. By skipping unwanted XML nodes, this avoids the need to always maintain a full copy of the XML document locally as processing of the document is performed remotely. When only partial views of XML documents are maintained, the processing requirements of mobile devices are less demanding and requires less memory. Furthermore, time and money can be saved when using mobile devices in bandwidth limited environments where data is often charged per kilobyte as only the relevant data is retrieved when the user selects the next node or branch.

Through extension of RXPP, a two-way exchange of XML documents is introduced called RXEP. RXEP allows users to receive XML fragments and also update remote XML documents. In addition to the navigation features of RXPP, RXEP further allows users to construct queries (e.g., using the XPath language), requesting many XML nodes from a remote XML document. In some cases, users can construct well crafted queries to retrieve all the relevant XML fragments using only a single request. RXEP locators are introduced which extend the path features of XPath to the provide precise location of received XML fragments within the clients own local version. RXEP locators provide extra information such as the nodes
absolute location and total number of sibling nodes. RXEP locators thus allow clients to retrieve fragments of XML whilst replicating the exact structure of the original XML document. Through exploitation of RXEP locators and RXEP’s two-way exchange, office suites using XML as a document format (such as MS Office and Openoffice), becomes an ideal target for collaborative editing amongst many users. This allows users to download only relevant parts of a document and upload corrections or modifications without the need to upload the entire document.

To further increase the efficiency of RXEP, a binarised (i.e., compressed) version of the protocol is explored. By utilising well established tree-based binarisation techniques significant savings can be achieved through compression of the RXEP structure and requested XML data. A new technique called SDOM is introduced which merges the structural information from XML Schemas with the requested XML document. SDOM allows users to request XML fragments using RXEP techniques where the requested XML data can be compressed on-the-fly using the information contained within SDOM. BinRXEP thus allows users to perform queries or navigation on remote XML documents and receive the results in a compact and compressed form. In many cases, the overhead added by RXEP, is reduced to less than a byte when using binRXEP.

Techniques for the transmission of both XML and XML Schema fragments within a single RXEP packet are proposed. Utilising RXEP, a user can request fragments with a of XML data from a remote document with a further option to request the XML Schema fragment required for validation of that fragment. In this way, the user can avoid retrieving all XML Schemas associated with an XML document, and may only retrieve the relevant XML Schema fragments.

Finally, the collaborative creation of XML Schemas is introduced. Utilising RXEP XML and Schema techniques, users can all contribute to the creation of a schema in realtime, while seeing the progress of other users. This collaborative creation of schemas can lead to quicker creation of XML Schemas. Users may then extend the current set of descriptors or generate new descriptors using ideas from the previous schema updates, thus resulting in a richer set of descriptors.
Statement of Originality

This is to certify that the work described in this thesis is entirely my own, except where due reference is made in the text.

No work in this thesis has been submitted for a degree to any other university or institution.

Signed

Stephen James Davis
18th of February, 2007
Acknowledgments

I would like to thank my supervisor Associate Professor Ian Burnett for his guidance, encouragement and support.

Thanks to my parents, Robert and Evelyn Davis, for their support and sacrifices over the years to get me here. Thanks to Eva Cheng, Chris Davis and Daniel Franklin for proof reading chapters of this thesis.

Finally, I would also like to thank my friends and colleagues of Whisper, SECTE and TITR for their support and enjoyable memories.
# Contents

## 1 Introduction

1.1 Thesis Outline ........................................... 3
1.2 Contributions .......................................... 4
1.3 Publications ........................................... 5
   1.3.1 Conference Publications .......................... 5
   1.3.2 Journal Publications .............................. 6
   1.3.3 Patents ........................................... 6
   1.3.4 MPEG ............................................ 6
   1.3.5 MPEG Contributions .............................. 6

## 2 Literature Review

2.1 Introduction ............................................. 8
2.2 XML and XML Tools ..................................... 8
   2.2.1 Extensible Markup Language ....................... 8
   2.2.2 XML Schema Languages ............................ 12
   2.2.3 XML Parsing Techniques ......................... 23
   2.2.4 XPath .......................................... 28
   2.2.5 MPEG-21 ....................................... 30
2.3 XML Transmission Techniques ........................... 36
   2.3.1 XML Fragment Interchange ....................... 36
   2.3.2 XML-Binary Optimized Packaging ............... 38
2.3.3 XStream ......................................................... 39
2.3.4 HyperText Transfer Protocol ................................. 42
2.3.5 File Transfer Protocol ........................................ 43
2.3.6 MPEG-7 TeM .................................................. 44
2.3.7 Fragment Caching for Mobile Devices ...................... 45
2.3.8 Delivery in Mobile Environments .............................. 46
2.4 XML Compression Techniques ................................. 47
  2.4.1 Redundancy Techniques ..................................... 49
  2.4.2 XML Conscious / Schema Based Compression Techniques ... 50
  2.4.3 Hybrid Techniques .......................................... 54
2.5 Conclusion ...................................................... 57

3 A New XML Delivery Protocol ................................. 61
  3.1 Introduction .................................................. 61
  3.2 A Remote XML Pull Protocol .................................. 61
    3.2.1 Requirements ............................................ 62
    3.2.2 RXPP Requests .......................................... 64
    3.2.3 XML Fragmentation ...................................... 65
    3.2.4 Node Identification in the Remote XML Document ........ 66
    3.2.5 Remote XML Navigation using RXPP ...................... 66
    3.2.6 Node by Node Navigation ................................ 67
    3.2.7 Level by Level Navigation ............................... 67
  3.3 The Remote XML Pull Protocol ............................... 69
    3.3.1 Invoking an RXPP Connection ............................ 69
    3.3.2 RXPP Commands ......................................... 70
    3.3.3 RXPP Scalability ........................................ 76
  3.4 Usage Scenarios .............................................. 77
### Contents

3.4.1 Scenario One: Navigation ........................................ 77  
3.4.2 Scenario Two: Random Access .................................. 81  
3.4.3 Scenario Three: Every Element is Retrieved .................. 81  
3.5 Conclusion ............................................................ 82  

4 Remote Exchange of XML Documents .............................. 84  
4.1 Introduction ............................................................ 84  
4.2 Remote XML Exchange Protocol ................................... 85  
4.3 RXEP Packets .......................................................... 89  
4.3.1 RXEP Request Packets ............................................. 90  
4.3.2 RXEP Response Packets .......................................... 97  
4.3.3 RXEP XPath Locators .............................................. 104  
4.4 RXEP XML Fragmentation Strategies ............................ 108  
4.4.1 Fragmentation by Navigation .................................... 108  
4.4.2 Fragmentation by Queries ........................................ 109  
4.4.3 Fragmentation by Navigation And Queries .................... 111  
4.5 Delivery of RXEP Messages .......................................... 112  
4.6 RXEP Experimental Results ......................................... 112  
4.6.1 Scenario One: RXEP on a Mobile Device .................... 115  
4.6.2 Scenario Two: Navigation ........................................ 116  
4.6.3 RXEP Scalability .................................................... 119  
4.7 Collaborative Editing Using RXEP ................................. 120  
4.7.1 Experimental Results .............................................. 121  
4.8 Creating a Standardised FRU Solution of RXEP ................. 124  
4.8.1 MPEG-B Requirements ............................................. 125  
4.8.2 MPEG-B Fragment Request Units ............................... 126  
4.8.3 Fragment Request Unit Syntax ................................... 126
5 XML Compression and the Binary RXEP Protocol

5.1 Introduction

5.2 XML Compression

5.2.1 Comparison of Lossless XML Compression Techniques

5.2.2 MPEG-B BiM in-Depth

5.3 Binarisation of MPEG-21 DIDs

5.3.1 Embedded Binary Data in MPEG-21 DIDs

5.3.2 Embedded XML in MPEG-21 DIDs

5.3.3 BiM DID Extension

5.4 Binary RXEP (BinRXEP)

5.4.1 Schema DOM Tree

5.4.2 BinRXEP Algorithm

5.4.3 Binary RXEP XPath Locators

5.4.4 Navigation with BinRXEP and encoded RXEP XPath Locators

5.4.5 Navigation with BinRXEP XML-Pull

5.4.6 BinRXEP Fragments from Queries

5.4.7 BinRXEP Experimental Results

5.5 Collaborative Editing with BinRXEP

5.5.1 Receive XML Documents with RXEP

5.5.2 Binarisation of Office XML Documents

5.5.3 Improving the Compression of WordML

5.6 Conclusion

6 XML Schema Exchange

6.1 Introduction

6.2 XML Schema Fragmentation
<table>
<thead>
<tr>
<th>CONTENTS</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>6.2.1 Extending RXEP for XML Schema Fragmentation</td>
<td>179</td>
</tr>
<tr>
<td>6.2.2 Client Requested XML Schema Fragments</td>
<td>182</td>
</tr>
<tr>
<td>6.2.3 Server Determined XML Schema Fragments</td>
<td>184</td>
</tr>
<tr>
<td>6.2.4 Combination of Server Determined and Client Requested</td>
<td>185</td>
</tr>
<tr>
<td>6.3 RXEP Schema XPath Locators</td>
<td>187</td>
</tr>
<tr>
<td>6.3.1 Experimental Results</td>
<td>188</td>
</tr>
<tr>
<td>6.4 XML Schema Generation</td>
<td>191</td>
</tr>
<tr>
<td>6.4.1 Collaborative XML Schema Generation</td>
<td>193</td>
</tr>
<tr>
<td>6.4.2 Experimental Results</td>
<td>194</td>
</tr>
<tr>
<td>6.5 Conclusion</td>
<td>198</td>
</tr>
<tr>
<td>7 Conclusions and Future Work</td>
<td>201</td>
</tr>
<tr>
<td>7.1 Conclusions</td>
<td>201</td>
</tr>
<tr>
<td>7.2 Future Work</td>
<td>204</td>
</tr>
<tr>
<td>Bibliography</td>
<td>207</td>
</tr>
<tr>
<td>A Software Implementation</td>
<td>215</td>
</tr>
<tr>
<td>A.1 Introduction</td>
<td>215</td>
</tr>
<tr>
<td>A.2 Implementation Details</td>
<td>216</td>
</tr>
<tr>
<td>A.2.1 SDOM Library</td>
<td>216</td>
</tr>
<tr>
<td>A.2.2 Binarisation Library</td>
<td>216</td>
</tr>
<tr>
<td>A.2.3 RXPP and RXEP Server and Client</td>
<td>216</td>
</tr>
<tr>
<td>A.2.4 Collaborative Schema Software</td>
<td>217</td>
</tr>
<tr>
<td>B RXEP Schema</td>
<td>221</td>
</tr>
<tr>
<td>C Thesis Files</td>
<td>224</td>
</tr>
</tbody>
</table>
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Example format for storing the a Personnel record in a plain text format</td>
<td>9</td>
</tr>
<tr>
<td>2.2</td>
<td>Example format for storing the Personnel record in a binary format</td>
<td>9</td>
</tr>
<tr>
<td>2.3</td>
<td>Example format for storing the Personnel record in a XML format</td>
<td>10</td>
</tr>
<tr>
<td>2.4</td>
<td>Extra metadata added to the Personnel record from Figure 2.3</td>
<td>10</td>
</tr>
<tr>
<td>2.5</td>
<td>Example of adding a photo to the Personnel record from Figure 2.4</td>
<td>11</td>
</tr>
<tr>
<td>2.6</td>
<td>The XML from Figure 2.5 illustrated graphically as tree structure, where</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>Elements are represented as ellipses and lines joining ellipses representing</td>
<td></td>
</tr>
<tr>
<td></td>
<td>parent/child relationship</td>
<td></td>
</tr>
<tr>
<td>2.7</td>
<td>Block diagram illustrating the process to determine if an XML document is</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>valid with respect to a schema (adapted from [12])</td>
<td></td>
</tr>
<tr>
<td>2.8</td>
<td>Example DTD to describe the personnel records</td>
<td>15</td>
</tr>
<tr>
<td>2.9</td>
<td>Example Relax NG schema to describe personnel records</td>
<td>16</td>
</tr>
<tr>
<td>2.10</td>
<td>Example of an XML Schema describing the Personnel Records</td>
<td>18</td>
</tr>
<tr>
<td>2.11</td>
<td>Example XML Schema containing a nested choices and sequences</td>
<td>19</td>
</tr>
<tr>
<td>2.12</td>
<td>Example DSD Schema describing schema for the Personnel Records</td>
<td>21</td>
</tr>
<tr>
<td>2.13</td>
<td>Example XML for three books with a name, author and rank</td>
<td>24</td>
</tr>
<tr>
<td>2.14</td>
<td>A DOM representation of the XML from Figure 2.13 illustrated graphically,</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>where the current node is represented by a bold-lined box</td>
<td></td>
</tr>
<tr>
<td>2.15</td>
<td>Example SAX events generated from parsing the XML in Figure 2.13, where</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>ellipsis represents removed output</td>
<td></td>
</tr>
<tr>
<td>2.16</td>
<td>Example XML-Pull events generated from calling the next() method when</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>parsing the XML in Figure 2.13, where ellipsis represents removed output</td>
<td></td>
</tr>
<tr>
<td>2.17</td>
<td>Example XML document defining a collection of books</td>
<td>29</td>
</tr>
</tbody>
</table>
2.18 Nodes as indicated in **bold** are the same nodes selected using two different XPath expressions //Book and /Collection//Book .......... 29

2.19 Partial view of a Digital Item Declaration representing a selection of songs of a classical music album. The DID provides the user with the choice of listening to the preview track or the full track .............. 33

2.20 Relationship of the principle elements within the Digital Item Declaration Model [30] ................................................................. 34

2.21 Example of adding additional MPEG-7 descriptors to a music track from the classical music DID from Figure 2.19 ......................... 35

2.22 Example XML document defining a collection of books .................. 37

2.23 Example Fragment using XML Fragment Interchange ......................... 37

2.24 Architecture of the XOP framework [35] ................................. 38

2.25 Example XML document with embedded photo ............................. 39

2.26 Example XML serialised as an XOP Package ............................... 40

2.27 Interaction diagram showing the operational flow of XStream (from [33] ...) 41

2.28 Format of the HTTP 1.1 Request format (from [37]) ......................... 42

2.29 Format of an HTTP 1.1 Response (from [37]) ................................. 42

2.30 Example of an HTTP 1.1 Request ................................................ 42

2.31 Example of an HTTP 1.1 Response to the Request in Figure 2.30 .......... 43

2.32 Model for FTP Use (from [38]) .................................................. 44

2.33 Different Kinds of compression and compression level [49] ............... 50

2.34 Example of XML Schema fragment and assigning binary codes based on the element position .................................................. 51

2.35 Simple XML document and corresponding tokenisation assignment (from [62]) 52

2.36 Example of the output stream from the Example XML in Figure 2.35 (from [62]) ................................................................. 52

2.37 Architecture of the Millau Compression - Decompression System [65] ... 53

2.38 Architecture of XMill Compressor [46] ......................................... 55

2.39 Architecture of the XGrind Compressor [58] .................................. 58
3.1 Example block diagram of the proposed system, illustrating the communication between the server and the client to exchange XML documents .......................... 62
3.2 RXPP field definition for and RXPP request ............................................ 64
3.3 Example of breaking a tree-like structure into two smaller fragments, which when added together, form the original structure ............................................. 66
3.4 Example of a Client beginning at t0 requesting the ‘next’ node. The server moves the current pointer to the Node c and delivers this node to the client. At time t3 the client adds this to its local version ............................................. 68
3.5 Example of a Client beginning at t0 issuing an expand command on the ‘current’ node c. The server moves the pointer to d, the first child of c, and sends these to the client. At t3 the client adds the node to its local version ............................................. 68
3.6 Example of a Client beginning at t0 requesting all children of the node c using an XPath locator. The server locates all children of c and sends these to the client. At t3 the client adds the children to its local version ............................................. 69
3.7 Example XML Document represented in a tree structure where the XML nodes represented as circles with the node name inside it ............................................. 72
3.8 Example fragment received by the the client after issuing a level depth of -1 on node C from Figure 3.7 ................................................................. 72
3.9 Example fragment received by the client after issuing a level depth of 1 on node A from Figure 3.7 ................................................................. 72
3.10 Example fragment received by the client after issuing a level depth of 2 on the root node A from Figure 3.7 ................................................................. 73
3.11 Example timing diagram illustrating the difference between Open and Closed modes of maintaining a server connection. a) illustrates the requests and responses when in closed mode. b) illustrates the requests and responses when in open mode ............................................. 74
3.12 An example book XML Schema, where the ellipsis represents removed XML Schema ................................................................. 76
3.13 Example Screenshot of client application ................................................................. 78
3.14 A sample portion of a single MP3 entry as used in the catalog when stored in the DID ................................................................. 79
3.15 Protocol Messaging Example ................................................................. 80
3.16 A sample portion of a MP3 list from the DID ................................................................. 81
LIST OF FIGURES

4.1 An example XML Schema which defines the Test type which contains a sequence of two elements ........................................... 87

4.2 An example XML Schema creating a new element myNewTest extending the Test type from the XML Schema in Figure 4.1 by adding an additional element to the sequence ........................................... 88

4.3 An example XML fragment illustrating the nodes required in MPEG-7 TeM to receive only the G element ........................................... 88

4.4 Example XML format of book1.xml located on an RXEP enabled server ................................................................. 89

4.5 Root XML Schema syntax for used for creating RXEP packets ................................................................. 90

4.6 RXEP request syntax for the RXEP schema used for creating RXEP request packets ................................................................. 91

4.7 RXEP XML Schema syntax for the Src type ................................................................. 91

4.8 Example RXEP request packet specifying to open an RXEP connection in open mode and specifying the source of /examples/xml/book1.xml ................................................................. 92

4.9 XML Schema syntax for the RXEP Query type ................................................................. 93

4.10 Example RXEP request packet specifying two RXEP query commands to request all Section nodes and its contents (i.e. levelDepth of -1) where the title is ‘Introduction’ as well as all the Section nodes for all Chapters containing an Introduction section, where results from both commands will be contained within a single RXEP response ................................................................. 93

4.11 XML Schema syntax for the RXEP XML-Pull type ................................................................. 95

4.12 Example RXEP request packet specifying an RXEP XMLPull command to return the ‘next’ node from the current node ................................................................. 95

4.13 Example RXEP request packet specifying multiple RXEP XMLPull commands, specifying the nodes to be returned are the ‘next’ node, followed by the new ‘next’ node, and then followed by the ‘expand’ command ................................................................. 96

4.14 XML Schema syntax for the RXEP Stream type ................................................................. 96

4.15 Example RXEP request packet specifying a RXEP Stream command, using the location attribute to instruct the sender that the starting point for streaming the XML fragments is the first chapter of a book specified by the XPath /Book/Chapter[1] ................................................................. 97

4.16 RXEP XML Schema syntax for the RXEP responseType used for creating RXEP response packets ................................................................. 98

4.17 Example client side XML document after navigation of the remote XML document in Figure 4.4 ................................................................. 98
4.18 RXEP Add response declaration for the RXEP schema used to notify to the receiver that the contained XML fragment is to be added to the location as defined in the location attribute ................................................................. 98

4.19 Example of RXEP Add response instructing the client to add the contents of Chapter 1 to the /Book/Chapter[1] ................................................................. 99

4.20 Updated XML document after the RXEP Add response from Figure 4.19 ................................ 99

4.21 RXEP Delete response declaration for the RXEP schema used to notify to the receiver that the XML node at the location as defined in the location attribute is to be deleted ......................................................... 100

4.22 Example of RXEP Delete response instructing the client to delete the second chapter ................................................................................................. 100

4.23 Updated XML document after the RXEP Delete response from Figure 4.22 ................................ 100

4.24 RXEP Update response declaration for the RXEP schema used to notify to the receiver that the location as defined in the location attribute is to be updated with the XML fragment contained within the RXEP response .................................................. 101

4.25 Example of RXEP Update response instructing the client to update the first chapter ................................................................................................. 101

4.26 Updated XML document after the RXEP Update response from Figure 4.25 ................................ 101

4.27 RXEP Insert response declaration for the RXEP schema used to notify to the receiver that the contained XML fragment is to be inserted as defined in the location and insertBefore attributes .................................................. 102

4.28 Example of RXEP Insert response, instructing the client to insert the new node after the node specified by the XPath /Book/Chapter[1] ................................ 102

4.29 Updated XML document after the RXEP Update response from Figure 4.25 ................................ 103

4.30 XML Schema syntax for the RXEPConfig type .............................................................................. 103

4.31 Example XML document containing a prefix definition on a node other than the root node ................................................................................................. 104

4.32 Example RXEP response with defining an RXEPConfig ................................................................................. 104

4.33 An example XML document .............................................................................................................. 104

4.34 An Example RXEP response resulting from a query /a/p/t[2] ................................................................................. 105

4.35 Two examples of valid XML documents which could be interpreted from the XPath expression /a/p/t[2] even though they have a different order ......................................................................................... 105

4.36 An Example RXEP request to retrieve the child nodes of /a and /a/p ................................................................................. 106
4.37 An Example RXEP response to the request in Figure 4.36, instructing placement of the XML child nodes to the parent nodes /a and /a/p: 107

4.38 Example of a local version of an XML document, resulting from the query /a/p/t[2]: 108

4.39 Example RXEP response using the RXEP XPath locator of /a/p[2,2]/[3,3], to provide the client with precise node placement: 108

4.40 An example of a remote XML document containing a library of books (only the Name elements are shown in this example, missing data is represented by ellipsis): 110

4.41 Example RXEP request looking for all Book nodes where the Name contains the string 'book': 110

4.42 Example RXEP response from the RXEP request in Figure 4.41: 110

4.43 An example of a local version of an XML document after navigation has revealed the format of the first track node: 112

4.44 Block diagram illustrating communication between two RXEP enabled devices: 113

4.45 Example RXEP request asking for all music tracks using HTTP as the transport mechanism: 113

4.46 Example RXEP response to the RXEP request from Figure 4.45 using HTTP as the transport mechanism: 114

4.47 Example RXEP request encapsulated within a SOAP message delivered via HTTP: 114

4.48 Example RXEP response to the RXEP request in Figure 4.47, encapsulated within a SOAP message delivered via HTTP: 115

4.49 Example RXEP Query request to query a remote baseball statistics XML document: 116

4.50 Example RXEP Response from the RXEP Query in Figure 4.49. For brevity, removed data is represented by ellipsis: 117

4.51 Example RXEP Response containing results from a query. Repeated element names are indicated by ellipsis: 118

4.52 Comparison of data uploaded and downloaded for the two tests: 122

4.53 RXEP comparison between the collaborative editing tests on two documents using OpenOffice and Microsoft XML documents: 123

4.54 Modified MPEG-7 Systems [7] diagram to accommodate the addition of FRUs: 127

4.55 MPEG-B Fragment Request Unit Syntax described using XML Schema: 127
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.56</td>
<td>Example MPEG-B Fragment Request Unit</td>
<td>127</td>
</tr>
<tr>
<td>4.57</td>
<td>Example Fragment Update Unit in response to the Fragment Request Unit in Figure 4.56</td>
<td>128</td>
</tr>
<tr>
<td>5.1</td>
<td>Example repetition within XML</td>
<td>132</td>
</tr>
<tr>
<td>5.2</td>
<td>Example BiM Syntax Tree generated from the MPEG-21 DIDL XML Schema</td>
<td>136</td>
</tr>
<tr>
<td>5.3</td>
<td>Example fragment of the DIDL XML Schema (shown graphically)</td>
<td>136</td>
</tr>
<tr>
<td>5.4</td>
<td>Example of a simple DID (valid to DIDL XML Schema) which declares a reference to a file on the local disk</td>
<td>137</td>
</tr>
<tr>
<td>5.5</td>
<td>Breakdown of the BiM Encoding technique</td>
<td>138</td>
</tr>
<tr>
<td>5.6</td>
<td>Example DID containing a description of a JPEG image, as well as the photo contained as an embedded resource, encoded as base64</td>
<td>141</td>
</tr>
<tr>
<td>5.7</td>
<td>An example of an MPEG-7 descriptor embedded in a MPEG-21 DID</td>
<td>144</td>
</tr>
<tr>
<td>5.8</td>
<td>Example of a bitstream generated using the MPEG-21 BiM extension when applied to an embedded XML which is valid to an XML Schema</td>
<td>145</td>
</tr>
<tr>
<td>5.9</td>
<td>Example of a resulting bitstream using the MPEG-21 BiM extension applied to an embedded XML which is not valid to an XML Schema (where the XML-aware compressor is defined in the decoderInit)</td>
<td>146</td>
</tr>
<tr>
<td>5.10</td>
<td>Example of a resulting bitstream using the MPEG-21 BiM extension applied to embedded XML which is not valid to an XML Schema (i.e., using a Text Compressor)</td>
<td>146</td>
</tr>
<tr>
<td>5.11</td>
<td>Example of assigning binary codes to the RXEP XML Schema</td>
<td>147</td>
</tr>
<tr>
<td>5.12</td>
<td>An example RXEP Request</td>
<td>147</td>
</tr>
<tr>
<td>5.13</td>
<td>An example Schema, presented as a tree view</td>
<td>150</td>
</tr>
<tr>
<td>5.14</td>
<td>Valid XML accordind to the Schema in Figure 5.13</td>
<td>150</td>
</tr>
<tr>
<td>5.15</td>
<td>SDOM representation of the XML in Figure 5.14 combined with the XML Schema information from Figure 5.13</td>
<td>151</td>
</tr>
<tr>
<td>5.16</td>
<td>Block diagram of the Binary RXEP system</td>
<td>152</td>
</tr>
<tr>
<td>5.17</td>
<td>Flowchart for encoding an XML Node</td>
<td>153</td>
</tr>
<tr>
<td>5.18</td>
<td>Flowchart for encoding an Element XML Node</td>
<td>154</td>
</tr>
<tr>
<td>5.19</td>
<td>Flowchart for encoding a complexType XML Node</td>
<td>155</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>5.20</td>
<td>Flowchart illustrating the encoding process of an RXEP XPath Locator</td>
<td></td>
</tr>
<tr>
<td>5.21</td>
<td>Tree view of an example XML Schema describing the format for a collection of media</td>
<td></td>
</tr>
<tr>
<td>5.22</td>
<td>Example XML document valid to the Schema shown in Figure 5.21</td>
<td></td>
</tr>
<tr>
<td>5.23</td>
<td>Example RXEP packet</td>
<td></td>
</tr>
<tr>
<td>5.24</td>
<td>The RXEP XML Schema illustrated as a tree-like structure, where the minOccurs and MaxOccurs are denoted using parentheses (,), and the nodes corresponding binary code for each node shown in bold font</td>
<td></td>
</tr>
<tr>
<td>5.25</td>
<td>SDOM representation of an RXEP request illustrating the binary output for each node in bold font. The total size of the binary output is 161 bits</td>
<td></td>
</tr>
<tr>
<td>5.26</td>
<td>Example RXEP response illustrating the binary codes (as bold font). The total binary output is 39 bits</td>
<td></td>
</tr>
<tr>
<td>5.27</td>
<td>Example RXEP XML-Pull Request illustrating the binary codes (as bold font). The total binary output is 13 bits</td>
<td></td>
</tr>
<tr>
<td>5.28</td>
<td>Example timing diagram illustrating the exchange of the binary codes using binRXEP to navigate through an XML document</td>
<td></td>
</tr>
<tr>
<td>5.29</td>
<td>Example binRXEP request where binary codes are illustrated in bold font</td>
<td></td>
</tr>
<tr>
<td>5.30</td>
<td>Comparison of upload and download of the test files using both binRXEP and RXEP</td>
<td></td>
</tr>
<tr>
<td>5.31</td>
<td>Comparison of text and binary compression</td>
<td></td>
</tr>
<tr>
<td>5.32</td>
<td>Sample of a Wordprocessing ML document</td>
<td></td>
</tr>
<tr>
<td>5.33</td>
<td>Relevant portion of the WordprocessingML Schemas showing the schema for the section elements</td>
<td></td>
</tr>
<tr>
<td>6.1</td>
<td>Example XML Schema for a Simple Media Album XML</td>
<td></td>
</tr>
<tr>
<td>6.2</td>
<td>Example XML document, valid to the XML Schema as shown in Figure 6.1</td>
<td></td>
</tr>
<tr>
<td>6.3</td>
<td>Modified RXEP requestType to provide the new SchemaQuery command to the RXEP XML Schema</td>
<td></td>
</tr>
<tr>
<td>6.4</td>
<td>The new schemaQueryType command to the RXEP XML Schema</td>
<td></td>
</tr>
<tr>
<td>6.5</td>
<td>Block diagram illustrating how clients can request XML fragments as well as XML Schema fragments</td>
<td></td>
</tr>
<tr>
<td>6.6</td>
<td>Example of an RXEP request to ask the server for the the XML Schema required for a specified XML Node</td>
<td></td>
</tr>
</tbody>
</table>
6.7 Example RXEP response containing the XML Schema fragment in response to the RXEP request in Figure 6.6. All ancestor schema information back to the root node is also sent to the client. 184

6.8 Block diagram illustrating how many XML Schema fragments (selected by the server) and the requested XML fragment can all be contained within a single RXEP response. 185

6.9 Example of an RXEP request specifying that the user wishes to receive both the XML Fragment and the corresponding XML Schema. 186

6.10 Example of a fragment of XML and corresponding XML Schema contained within a single RXEP packet, as a result from the RXEP request from Figure 6.9. 186

6.11 Example reduction of an RXEP XPath locator when applying the RXEP Schema XPath simplifications. 189

6.12 Screenshot of the client application’s representation of the fragments of XML Schema received. 190

6.13 Portion of the DID XML document received by the user, where the song title of the track is missing. 191

6.14 Screenshot of the client JAVA application used for the schema experiment. 194

6.15 Initial XML Schema used as the starting point for users to add their descriptors to. This XML Schema specified a root element of Album which a choice of two child nodes, Videos and Photos. 195

6.16 Output from the collaboratively edited XML Schema experiment created by five people using the RXEP techniques. Note that this is not a valid XML Schema as the complexType and choice elements have been removed for ease of viewing for the people in the experiment. 198

6.17 Illustrates the number of updates per user during the XML Schema experiment. 199

6.18 Illustrates the number of updates, over time, by each user during the XML Schema experiment. 199

6.19 Plot of the number of updates vs the level depth of the added nodes. 200

7.1 Block diagram illustrating. 205

A.1 Screenshot of the test binarisation encoder, showing the SDOM conversion of the input XML document. 217

A.2 Screenshot of the client RXEP JAVA application. 218
A.3  Screenshot of the collaborative schema client JAVA application . . . . . . . . 218
A.4  Screenshot of the collaborative schema server JAVA application . . . . . . . 219
A.5  Screenshot of the collaborative schema log analyser JAVA application . . . . 220
# List of Tables

2.1 Boolean expression constructs in DSD2 (From [12]) . . . . . . . . . . . . . . 22  
2.2 Regular expression constructs in DSD2 (From [12]) . . . . . . . . . . . . . . 22  
2.3 Commonly used functions from the DOM 3 Node Interface . . . . . . . . . . . 25  
3.1 RXPP Parsing Times and Memory Requirements . . . . . . . . . . . . . . . . 77  
3.2 Comparison of RXPP with other technologies . . . . . . . . . . . . . . . . . 83  
4.1 RXEP Parsing Times and Memory Requirements . . . . . . . . . . . . . . . . 119  
4.2 Comparison of RXEP with other technologies . . . . . . . . . . . . . . . . . 129  
5.1 Comparison of compression results on test files 1 - 13 (all units in Bytes) . . 134  
5.2 Comparison of RXEP, binRXEP and Zip for scenario one (All units in bytes) 167  
5.3 Comparison of RXEP, binRXEP and Zip for scenario two (All units in bytes) 168  
5.4 Comparison of BinRXEP with other technologies . . . . . . . . . . . . . . . . 175
## List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>ASCII</td>
<td>American Standard Code for Information Interchange</td>
</tr>
<tr>
<td>ASN.1</td>
<td>Abstract Syntax Notation number One</td>
</tr>
<tr>
<td>AU</td>
<td>Access Unit</td>
</tr>
<tr>
<td>BinRXEP</td>
<td>Binary Remote XML Exchange Protocol</td>
</tr>
<tr>
<td>BiM</td>
<td>Binary format for MPEG-7</td>
</tr>
<tr>
<td>CODEC</td>
<td>enCOder / DECoder</td>
</tr>
<tr>
<td>DI</td>
<td>Digital Item</td>
</tr>
<tr>
<td>DID</td>
<td>Digital Item Declaration</td>
</tr>
<tr>
<td>DOM</td>
<td>Document Object Model</td>
</tr>
<tr>
<td>DTD</td>
<td>Document Type Definition</td>
</tr>
<tr>
<td>DIDL</td>
<td>Digital Item Declaration Language</td>
</tr>
<tr>
<td>FTP</td>
<td>File Transfer Protocol</td>
</tr>
<tr>
<td>FRU</td>
<td>Fragment Request Unit</td>
</tr>
<tr>
<td>FUU</td>
<td>Fragment Update Unit</td>
</tr>
<tr>
<td>GPRS</td>
<td>General Packet Radio Service</td>
</tr>
<tr>
<td>HTTP</td>
<td>HyperText Transport Protocol</td>
</tr>
<tr>
<td>IEC</td>
<td>International Electrotechnical Commission</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>ISO</td>
<td>International Organization for Standardization</td>
</tr>
<tr>
<td>MIME</td>
<td>Multipurpose Internet Mail Extensions</td>
</tr>
<tr>
<td>MPEG</td>
<td>Moving Pictures Expert Group</td>
</tr>
<tr>
<td>OWL</td>
<td>Web Ontology Language</td>
</tr>
</tbody>
</table>
List of Abbreviations

PSVI  Post Schema Validation Infoset
RAM  Random Access Memory
RXEP  Remote XML Exchange Protocol
RDF  Resource Description Framework
RXPP  Remote XML Pull Protocol
SAX  Simple API for XML
SDOM  Schema Document Object Model
SOAP  Simple Object Access Protocol
SQL  Structured Query Language
TeM  Textual Encoding format for MPEG-7
URI  Uniform Resource Identifier
URL  Uniform Resource Locator
VLC  Variable Length Coding
UTF-8  UCS Transformation Format 8
W3C  World Wide Web Consortium
WAP  Wireless Application Protocol
WSDL  Web Services Description Language
XML  eXtensible Markup Language
XOP  XML-Binary Optimized Packaging