

University of Wollongong
Research Online

University of Wollongong Thesis Collection
1954-2016

University of Wollongong Thesis Collections

2007

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>

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: This work is copyright. Apart from any use permitted under the Copyright Act 1968, no part of this work may be reproduced by any process, nor may any other exclusive right be exercised, without the permission of the author. 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.

Unless otherwise indicated, the views expressed in this thesis are those of the author and do not necessarily represent the views of the University of Wollongong.

Recommended Citation

Davis, Stephen James, Novel Approaches to the Delivery of XML and Schemas, PhD thesis, Electrical, Computer and Telecommunications Engineering, University of Wollongong, 2007. <http://ro.uow.edu.au/theses/682>

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
1.1	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	8
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

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

4.9	Conclusion	128
5	XML Compression and the Binary RXEP Protocol	130
5.1	Introduction	130
5.2	XML Compression	131
5.2.1	Comparison of Lossless XML Compression Techniques	132
5.2.2	MPEG-B BiM in-Depth	135
5.3	Binarisation of MPEG-21 DIDs	139
5.3.1	Embedded Binary Data in MPEG-21 DIDs	141
5.3.2	Embedded XML in MPEG-21 DIDs	142
5.3.3	BiM DID Extension	143
5.4	Binary RXEP (BinRXEP)	146
5.4.1	Schema DOM Tree	149
5.4.2	BinRXEP Algorithm	152
5.4.3	Binary RXEP XPath Locators	156
5.4.4	Navigation with BinRXEP and encoded RXEP XPath Locators	159
5.4.5	Navigation with BinRXEP XML-Pull	162
5.4.6	BinRXEP Fragments from Queries	165
5.4.7	BinRXEP Experimental Results	166
5.5	Collaborative Editing with BinRXEP	169
5.5.1	Receive XML Documents with RXEP	169
5.5.2	Binarisation of Office XML Documents	171
5.5.3	Improving the Compression of WordML	171
5.6	Conclusion	174
6	XML Schema Exchange	176
6.1	Introduction	176
6.2	XML Schema Fragmentation	177

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

List of Figures

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

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

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 <code>myNewTest</code> 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 <code>book1.xml</code> 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 <code>/examples/xml/book1.xml</code> .	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. <code>levelDepth</code> of -1) where the title is 'Introduction' as well as all the <code>Section</code> 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 <code>/Book/Chapter[1]</code>	97
4.16	RXEP XML Schema syntax for the RXEP <code>responseType</code> 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	105
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	106
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

4.56	Example MPEG-B Fragment Request Unit	127
4.57	Example Fragment Update Unit in response to the Fragment Request Unit in Figure 4.56	128
5.1	Example repetition within XML	132
5.2	Example BiM Syntax Tree generated from the MPEG-21 DIDL XML Schema	136
5.3	Example fragment of the DIDL XML Schema (shown graphically)	136
5.4	Example of a simple DID (valid to DIDL XML Schema) which declares a reference to a file on the local disk	137
5.5	Breakdown of the BiM Encoding technique	138
5.6	Example DID containing a description of a JPEG image, as well as the photo contained as an embedded resource, encoded as base64	141
5.7	An example of an MPEG-7 descriptor embedded in a MPEG-21 DID	144
5.8	Example of a bitstream generated using the MPEG-21 BiM extension when applied to an embedded XML which is valid to an XML Schema	145
5.9	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)	146
5.10	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)	146
5.11	Example of assigning binary codes to the RXEP XML Schema	147
5.12	An example RXEP Request	147
5.13	An example Schema, presented as a tree view	150
5.14	Valid XML accordind to the Schema in Figure 5.13	150
5.15	SDOM representation of the XML in Figure 5.14 combined with the XML Schema information from Figure 5.13	151
5.16	Block diagram of the Binary RXEP system	152
5.17	Flowchart for encoding an XML Node	153
5.18	Flowchart for encoding an Element XML Node	154
5.19	Flowchart for encoding a complexType XML Node	155

5.20	Flowchart illustrating the encoding process of an RXEP XPath Locator	157
5.21	Tree view of an example XML Schema describing the format for a collection of media	158
5.22	Example XML document valid to the Schema shown in Figure 5.21	158
5.23	Example RXEP packet	159
5.24	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	160
5.25	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	161
5.26	Example RXEP response illustrating the binary codes (as bold font). The total binary output is 39 bits	162
5.27	Example RXEP XML-Pull Request illustrating the binary codes (as bold font). The total binary output is 13 bits	163
5.28	Example timing diagram illustrating the exchange of the binary codes using binRXEP to navigate through an XML document	164
5.29	Example binRXEP request where binary codes are illustrated in bold font . .	167
5.30	Comparison of upload and download of the test files using both binRXEP and RXEP	170
5.31	Comparison of text and binary compression	170
5.32	Sample of a Wordprocessing ML document	172
5.33	Relevant portion of the WordprocessingML Schemas showing the schema for the section elements	173
6.1	Example XML Schema for a Simple Media Album XML	178
6.2	Example XML document, valid to the XML Schema as shown in Figure 6.1 .	179
6.3	Modified RXEP requestType to provide the new SchemaQuery command to the RXEP XML Schema	181
6.4	The new schemaQueryType command to the RXEP XML Schema	181
6.5	Block diagram illustrating how clients can request XML fragments as well as XML Schema fragments	183
6.6	Example of an RXEP request to ask the server for the the XML Schema required for a specified XML Node	184

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

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

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