- 2 Gerald Beuchelt, Robert Dingwell, Andrew Gregorowicz, Harry Sleeper
- 3 The MITRE Corporation
- 4 202 Burlington Rd.
- 5 Bedford, MA 01730
- 6 U.S.A.
- 7 © 2009 The MITRE Corporation. All rights reserved.

1 Introduction 8

- 9 The hData Record Format (HRF) describes the XML representation of the continuity of care information in an
- 10 electronic health record (EHR). The HRF is implemented through a component-specific XML documents that are
- 11 linked and organized through a "master document". For better organization, the individual XML documents are
- 12 put into a hierarchy, with the master document at the root of this hierarchy. While the HRF defines a core set
- 13 of components, it is fully extensible and can easily be adopted for more complex situations.
- 14 This specification only describes the organization of data within an abstract hData Record (HDR). Another
- 15 specification describes how a HDR is serialized [1].

16 1.1 Namespaces

- 17 This document uses the following namespaces. This specification uses a number of namespace prefixes
- throughout; they are listed in Table 1. Note that the choice of any namespace prefix is arbitrary and not 18
- 19 semantically significant.

Namespace Prefix	Namespace URI	Description
hrf	http://projecthdata.org/hdata/schemas/2009/06/core	Namespace for elements in this document
hrf-md	http://projecthdata.org/hdata/schemas/2009/11/metadata	Namespace for meta data
xs	http://www.w3.org/2001/XMLSchema	XML Schema namespace

20

21

1.2 Glossary (Non-Normative)

- 22 hData Record Format (HRF) - The part of the hData specification that defines the abstract hierarchy, meta-data 23
- schema, and document organization of the hData record.

- hData Record (HDR) an single instantiation of the HRF.
- 25 hData Restful API (HRA) the part of the hData specification that defines the basic HTTP-based API for
- accessing or modifying an HDR.
- 27 **hData Specification** a normative specification that defines the HRF, the HRA, and a file-based serialization
- 28 format.
- 29 hData Content Profile (HCP) a profile of the medical content of an HDR. An HCP is specified separately from
- 30 the HRF. The hData Project defines an initial HCP (iHCP) that covers the 35 data elements for EHRs/EMRs
- 31 defined by the National Quality Foundation.
- 32 **Electronic Medical Record (EMR)** the medical record or records of a single patient in the IT system of an actor
- 33 (health provider, government entity, payer, etc.). In this definition, an HDR is a type of EMR.
- 34 Electronic Health Record (EHR) the collection of all EMRs of a single patient, across organizational and
- 35 national boundaries.
- 36 **EHR System** An IT system that creates, stores, and manages EMRs.
- 37 Clinical Document Architecture (CDA) an XML specification by Health Layer 7 (HL7) that is intended to be
- 38 used for EMRs.
- 39 Continuity of Care Record (CCR) a specification by ASTM that is intended to be used for summary/continuity
- 40 of care documentation. A CCM is a type of EMR.
- 41 Continuity of Care Document (CCD) a profile of the CDA that accommodates the medical information of the
- 42 CCR.

- 43 HITSP/C32 (C32) a constrained profile of the CCD that is intended to simplify implementation and improve
- interoperability. There is no normative schema for C32. Note that HITSP has recently split up C32 into
- 45 HITSP/C80 and HITSP/C83.
- 46 MITRE/L32 (L32) a significantly constrained profile of the C32 specification. L32 comes with a normative
- 47 schema and can be mapped onto the HRF.
- 48 1.3 Notational Conventions
- The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT",
- 50 "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.
- 51 When describing concrete XML schemas, this specification uses the following notation: each member of an
- 52 element's [children] or [attributes] property is described using an XPath-like notation (e.g.,
- /x:MyHeader/x:SomeProperty/@value1). The use of {any} indicates the presence of an element wildcard. The
- use of @{any} indicates the presence of an attribute wildcard.
- Note also that only the W3C XML schemas linked in Appendix A at the end of this document are normative –
- 57 any schema fragment or other schema description is informational only.

58 2 Hierarchical Organization

- 59 The basic approach of the hData Record Format is to represent the Continuity of Care data through linked XML
- documents, which are organized through an abstract hierarchy. The hData storage and network protocols map
- 61 this abstract hierarchy to a concrete implementation, such as a directory folder or web resource hierarchy.
- 62 HRF only defines representations for the most common CCD constructs. In order to be able to accommodate
- 63 more complex situations, it was designed with a number of extension points that allow the definition and
- 64 insertion of new components. Extension can be optional, i.e. a parser that is not capable of processing the data
- in the extension may safely ignore it. If an extension is marked mandatory and the parser has no support for it,
- the parser MUST notify the user or raise an exception.

67 2.1 Overall Structure

- At the root of the hierarchy is the master document (MD) and additional documents, that have relevance to
- 69 the entirety of the continuity of care document. The actual information is contained in component specific
- sections, some of which are REQUIRED. The component specific sections are the primary extension points
- 71 within the hierarchy. Implementers can either extend existing component sections or define new sections.
- 72 Such newly created sections MUST be registered in the MD to be accessible.
- 73 Each section corresponds to a single set of XML documents, i.e. each section is associated with documents that
- comply to a given schema. Section documents that are required by this specification are defined through W3C
- 75 XML Schema.

76

79

80

81

82

83 84

85

86 87

88

89 90

91

92 93

94

95

2.2 Root Document

- 77 The root document is as the root of the hierarchy. It contains the following elements (REQUIRED if not marked otherwise):
 - /hrf:id This element uniquely identifies the document, e.g. through a textual representation of a UUID. It is RECOMMENDED to not use absolute URIs that may be assumed to be resolvable to a concrete resource location.
 - /hrf:version The version of the hData Record Format used within this document.
 - /hrf:created Creation date of the document, using the W3C XML Schema Date data type. This data SHOULD be significant to at least the second.
 - /hrf:lastModified Last modification of the document, using the W3C XML Schema Date data type. This data SHOULD be significant to at least the second.
 - /hrf:extensions Node containing a list of extensions (list of hrf:extension elements). Any extension to this specification MUST register itself in this section.
 - /hrf:extensions/hrf:extension (OPTIONAL) This element contains a unique identifier for the extension.
 - /hrf:extensions/hrf:extension/@requirement This attribute MUST be either "mandatory" or "optional". If the parser has no support for a mandatory extension, it MUST notify the user or raise an exception. It is RECOMMENDED to not process the document.
 - /hrf:sections This node contains references to all component-specific sections (hrf:section)
 - /hrf:sections/hrf:section (OPTIONAL) A hrf:section element MAY contain additional hrf:section elements. The path attributes SHOULD be concatenated to construct a hierarchy.

- /hrf:sections/hrf:section/@path This attribute is really a path segment, used to construct the full path to the section from the root.
 - /htf:sections/hrf:section/@typeId A unique identifier. This identifier MUST conform to the requirements for XML namespace identifiers. It represents the default content type for this section.
 - /hrf:sections/hrf:section/@name Used for a human-friendly name to this section.
- 101 Extensions MAY extend the master document with additional elements, such as e.g. a mechanism to record
- versions of the data contained in the document.
- **103 2.3 Sections**

98

99

100

118119

120121

122

123124

125126

127

128

129130

131132

133

- 104 Section within a hData record form an abstract hierarchy, similar to the file folder structure commonly used in
- 105 hierarchical file systems. Section can contain either Section Documents or other Sections. Sections are
- identified by their path. The path to a Section is constructed by starting with a forward slash ("/") and
- appending all section path names from the root of the HDR to the Section.
- 108 2.4 Section Documents
- 109 At each section a collection of documents can be obtained. Within each Section, there MUST NOT be more
- than one type of section documents, identified by the URI of the typeId attribute in the corresponding section
- 111 node of the root document. Any URI used within the sections node for the typeId attribute MUST be registered
- as an Extension in the extensions node of the root document.
- 113 2.4.1 Section Document Meta Data
- Each section contains a collection of meta data artifacts that are associated with each Section Document in an
- 115 XML fragment starting with <hrf-md:DocumentMetaData>
- /DocumentMetaData DocumentMetaData is the top-level element for the hData meta data
 specification.
 - /DocumentMetaData/@MediaType This attribute contains the media type of the document itself. If it is not present, the default media type of the content type is assumed.
 - /DocumentMetaData/@ContentType This attribute contains the URI for the content type of this
 document. If it is not present, the content type for the Section is implied. Note that the current hData
 Content Profiles assume that the content type for all Section Documents within a given Section is
 uniform.
 - /DocumentMetaData/PedigreeInfo (OPTIONAL) This optional node holds the pedigree information for the Section Document. It is of type <hrf-md:PedigreeInfo>
 - /DocumentMetaData/DocumentId This required element of type xs:String holds an identifier for the Section Document. It MUST be unique over any given Section.
 - /DocumentMetaData/LinkedDocuments (OPTIONAL) This optional node holds a list of URI links to documents that are related to this Section Document. Use depends on the semantics of the Section Document Type. It can have <hrf-md:LinkInfo> typed child elements.
 - /DocumentMetaData/RecordDate This required node holds the information about Document creation and modification.
 - /DocumentMetaData/ReecordDate/CreatedDateTime This required element of type <xs:dateTime>
 contains the dateTime of creation of this document. If this document is not derived (see

- PedigreeInfo), this is the time of the creation of the original. If this document is derived from another origin, this element contains the date of derivation.
 - /DocumentMetaData/ReecordDate/Modified (OPTIONAL) This optional node is first created when the
 document is changed for the first time. It contains a collection of modification dates with optional
 pedigree information of the modifier.
 - /DocumentMetaData/ReecordDate/Modified/ModfiedDateTime This required element of type <xs:dateTime> records a dateTime when the document was modified.
 - /DocumentMetaData/ReecordDate/Modified/PedigreeInfo (OPTIONAL) This optional node of type <hrf-md:PedigreeInfo> contains the pedigree information of the modifier.
 - /DocumentMetaData/Confidentiality (OPTIONAL) This element contains controls for confidentiality details are TBD.
 - /DocumentMetaData/AccessControl (OPTIONAL) This element contains controls for access control details are TBD.
 - /DocumentMetaData/Consent (OPTIONAL) This element contains controls for consent details are TBD.
- There are two more types that are being used in <DocumentMetaData>: <hrf-md:PedigreeInfo> and <hrf-
- md:LinkInfo>. This is the schema for <hrf-md:PedigreeInfo>
- /PedigreeInfo This node contains the pedigree information.
 - /PedigreeInfo/XmlSignature (OPTIONAL) This optional node contains the signature information on the document or this meta data.
 - /PedigreeInfo/XmlSignature/@documentMethod This optional attribute indicates what method was used to transform binary Section Document mediatypes into XML files for signature. Currently the only permitted methods are xml, sha256 and base64. xml is the default XML signature over XML documents. base64 encodes a data stream into an XML document. The root node it root and contains the BASE64 encoded data. sha256 calculates a hash over the binary stream and signs this hash.
 - /PedigreeInfo/XmlSignature/ds:Signature (0..unbounded) A collection of XML Signatures. This Signature MUST contain: 1. a valid Reference to either the metadata or the Section Document 2. the ds:KeyInfo for the signer (optional with DSig required here)
 - /PedigreeInfo/Source (OPTIONAL) This node indicates the source of this data.
 - /PedigreeInfo/Source/@derived If the data is derived (i.e. copied or compiled from other sources) this attribute of type <xs:boolean> MUST be set to true.
 - /PedigreeInfo/Source/PedigreeInfo (0..unbounded) This element contains the <hrf-md:PedigreeInfo> of the all source from which this document was derived.
 - /PedigreeInfo/Source/Document (0..unbounded) This element of type <hrf-md:LinkInfo> contains links to all documents from which this document was derived.
 - /PedigreeInfo/Author (0..unbounded) This element contains the names or identifiers of all author(s).
 - /PedigreeInfo/Organization (0..unbound) This element identified the organization(s) at which this document was created.
- 173 This is the schema for <hrf-md:LinkInfo>:
- 174 ◆ /LinkInfo

137

138

139

140141

142143

144

145

146147

148

149

153

154

155

156

157

158159

160161

162163

164165

166

167

168169

170171

- LinkInfo/Target –This required element of type <xs:anyURI> contains the absolute link to the
 referenced SectionDocument.
 - /LinkInfo/##any (OPTIONAL) extension point.

178 2.5 hData Content Profiles

- 179 This specification does not specify which sections are required for an hData Record. This is done in separate
- 180 hData Content Profiles (HCP).

181 3 Common Data Types

- 182 Common data types such as address, person information, etc. in section documents SHOULD use the data
- types described below. They are contained in the http://projecthdata.org/hdata/schemas/2009/06/core
- 184 schema.

177

- 185 **3.1 Name**
- 186 This element represents the name of a person. It contains the following elements:
- /hrf:name/hrf:title (OPTIONAL) The person's title, such as Mr., Dr., etc.
- /hrf:name/hrf:given Used to represent a person's given names. A person's first name SHOULD be
 present in the first occurrence of the given element. Middle names SHOULD appear in subsequent
 occurrences of the given element.
- /hrf:name/hrf:lastname Used to represent the person's surname or family name
 - /hrf:name/hrf:suffix (OPTIONAL) A suffix for the person's name, such as Jr., Sr., III, etc.
- **3.2 Address**

- 194 This element provides a representation of a postal address. It contains the following elements:
- /hrf:address/hrf:streetAddress (OPTIONAL) SHOULD contain one line of the postal address. This element MAY be repeated to capture multiple lines of a postal address. This element MUST NOT contain city, state, zip code or country information.
- /hrf:address/hrf:city The city of the postal address
- /hrf:address/hrf:stateOrProvince The state or province of the postal address. For US States, this value
 MUST be represented in FIPS State Alpha Code (http://www.itl.nist.gov/fipspubs/fip5-2.htm)
- /hrf:address/hrf:zip
- /hrf:address/hrf:country (OPTIONAL) The country of the postal address. If present, the country
 name MUST be represented as an ISO 3166-1 country name.
- **204 3.3 Telecom**
- Telecom elements are used to describe various forms of contact.

- /hrf:telecom/@use This attributes describes whether the contact is for an individual's residence,
 place of business , vacation home, or other.
 - /hrf:telecom/@value This attribute states the actual contact means and MUST be in url semantics.
 - /hrf:telecome/@preferred (OPTIONAL) Boolean attributes that denotes whether the telecom is a preferred means of contact.

212 **3.4 Person**

208209

210

211

214215216

217

218

219

226

229

236

237

238

- This element provides the representation of basic demographic information about an individual.
 - /hrf:person/name The name of the individual defined in an hrf:name element as described above.
 - /hrf:person/hrf:addresses A list of address information related to the individual defined using the hrf:address structure described above
 - /hrf:person/hrf:telecom 0 or more hrf:telecom elements related to the individual described using the hrf:telecom element described above.
- 220 **3.5** Actor
- The Actor substitutionGroup is used to represent situations when an entity may be either a person or an
- 222 organization.
- 223 3.6 Organiation
- This element is used to represent an organization and the basic deogrphic information associated with the
- 225 organization.
 - /hrf:organization/hrf:name The name of the organization, this is a simple string value
- /hrf:organization/hrf:pointsOfContact A list of 0 or more points of contact for the organization
 represented as a list of hrf:person elements described above.
 - /hrf:organization/hrf:address 0 or more hrf:address elements for the organization
- 230 3.7 CodedValue
- 231 This is not an element but rather a complexType for a generalized approach for creating elements that require
- coded value information. Leaving it as a complexType allows for the codedValue to have a more meaningful
- 233 name to the element derived from it while still retaining the generic codedValue attributes. As this is a
- complexType the xpath statements below are not accurate as hrf:codedValue would need to be replaced with
- an actual instantiation of the complexType
 - /hrf:codedValue/@code the code value from the codeSystem being used
 - /hrf:codedValue/@codeSystem the coded system from which the code is from
 - /hrf:codedValue/@version the version of the codeSystem used
- /hrf:codedValue/@displayName the displayName of the codedValue as described by the
 codeSystem
 - /hrf:codedValue/text() codedValue elements can contain a free text block to further describe the coded value element in question

243	3.8 Date This element is used to represent a singular point in time.	
244		
245	/hrf:date/text() – the value of the date in question in xsd:date format	
246	3.9 DateRange	
247	This element is used to represent a date range.	
248	 /hrf:dateRange/hrf:low – the low end of the date range represented as an hrf:date element 	
249 250	 /hrf:dateRange/@high - (Optional) this represents the high end of the date range represented as an hrf:date element or if not included represents an open ended date range 	
251	3.10 InformationSource	
252	This element is used to represent where the information in a section may have originated from.	
253 254	 /hrf:/informationSource/hrf:author – (Optional) the author of the referenced document represented as an hrf:person element 	
255	 /hrf:informationSource/date - (Optional) the date the referenced document was created 	
256	represented as an hrf:date element	
257 258	 /hrf:informationSource/reference – (Optional) a reference to the document from which the section information was derived 	
259	 /hrf:informationSource/informant - the individual or organization who added the information to 	
260	the record, represented as an hrf:actor as described above	
261	3.11 Description	
262		
263	information.	
264	/hrf:description/hrf:text – free text block	
265	 /hrf:description/hrf:codedValue – list of codedValues that pertain to the description 	
266	3.12 AbstractSection	
267	This abstract complex type is used to represent a set of common feature that all section documents should	
268	contain. As this is a complexType the xpath statements below are not accurate as hrf:abstractSection would	
269	need to be replaced with an actual instantiation of the complexType.	
270	 /hrf:abstractSection/hrf:description – (Optional) as described above 	
271	 /hrf:abstractSection/hrf:informationSource – (Optional) as described above 	
272	4 Appendix A: Normative Schemas	
273	4.1 Root Document	
27/	This section contains the normative schema for the root document (see Section 2.2)	

<?xml version="1.0" encoding="UTF-8"?>

```
276
     <!-- Copyright 2009 The MITRE Corporation
277
278
     Licensed under the Apache License, Version 2.0 (the "License");
279
     you may not use this file except in compliance with the License.
280
     You may obtain a copy of the License at
281
282
     http://www.apache.org/licenses/LICENSE-2.0
283
284
     Unless required by applicable law or agreed to in writing, software
285
     distributed under the License is distributed on an "AS IS" BASIS,
     WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
286
287
     implied.
288
     See the License for the specific language governing permissions and
289
     limitations under the License. -->
290
291
     <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"</pre>
292
     elementFormDefault="qualified"
293
     targetNamespace="http://projecthdata.org/hdata/schemas/2009/06/core"
294
     xmlns:core="http://projecthdata.org/hdata/schemas/2009/06/core">
295
       <xs:element name="root">
296
         <xs:complexType>
297
           <xs:all>
298
             <xs:element ref="core:documentId"/>
299
             <xs:element ref="core:version"/>
             <xs:element ref="core:created"/>
300
             <xs:element ref="core:lastModified"/>
301
             <xs:element ref="core:extensions"/>
302
303
             <xs:element ref="core:sections"/>
304
           </xs:all>
305
         </xs:complexType>
306
       </xs:element>
       <xs:element name="documentId" type="xs:string"/>
307
308
       <xs:element name="version" type="xs:string"/>
       <xs:element name="created" type="xs:date"/>
309
310
       <xs:element name="lastModified" type="xs:date"/>
       <xs:element name="extensions">
311
312
         <xs:complexType>
313
           <xs:sequence>
314
             <xs:element minOccurs="0" maxOccurs="unbounded"</pre>
     ref="core:extension"/>
315
316
           </xs:sequence>
         </xs:complexType>
317
318
       </xs:element>
319
       <xs:element name="extension">
         <xs:complexType mixed="true">
320
321
           <xs:attributeGroup ref="core:extension"/>
322
         </xs:complexType>
323
       </xs:element>
324
       <xs:element name="sections">
325
         <xs:complexType>
326
           <xs:sequence>
             <xs:element minOccurs="0" maxOccurs="unbounded"</pre>
327
328
     ref="core:section"/>
329
           </xs:sequence>
         </xs:complexType>
330
```

```
331
       </xs:element>
332
       <xs:attributeGroup name="extension">
333
         <xs:attribute name="requirement" use="required">
334
           <xs:simpleType>
335
              <xs:restriction base="xs:token">
336
                <xs:enumeration value="mandatory"/>
337
                <xs:enumeration value="optional"/>
338
              </xs:restriction>
339
           </xs:simpleType>
340
         </xs:attribute>
       </xs:attributeGroup>
341
342
       <xs:element name="section">
343
         <xs:complexType>
344
           <xs:sequence>
              <xs:element minOccurs="0" maxOccurs="unbounded"</pre>
345
346
     ref="core:section"/>
347
           </xs:sequence>
348
           <xs:attribute name="path" use="required"/>
349
           <xs:attribute name="name" use="required"/>
350
           <xs:attribute name="typeId" use="required"/>
351
         </xs:complexType>
352
       </xs:element>
353
     </xs:schema>
```

4.2 Section Document Meta Data

354

355

356

This section contains the normative schema for the Section Document meta data (see Section 2.4.1).

```
<?xml version="1.0" encoding="UTF-8"?>
357
358
     <!-- Copyright 2009 The MITRE Corporation
359
360
     Licensed under the Apache License, Version 2.0 (the "License");
361
     you may not use this file except in compliance with the License.
     You may obtain a copy of the License at
362
363
364
     http://www.apache.org/licenses/LICENSE-2.0
365
366
     Unless required by applicable law or agreed to in writing, software
367
     distributed under the License is distributed on an "AS IS" BASIS,
     WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
368
369
     implied.
370
     See the License for the specific language governing permissions and
371
     limitations under the License. -->
372
373
     <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"</pre>
374
         xmlns:hd-
375
     md="http://projecthdata.org/hdata/schemas/2009/11/metadata"
376
377
     targetNamespace="http://projecthdata.org/hdata/schemas/2009/11/metada
378
     ta">
379
         <xs:include</pre>
380
             schemaLocation="http://www.w3.org/TR/2002/REC-xmldsig-core-
381
     20020212/xmldsig-core-schema.xsd"/>
382
         <xs:complexType name="DocumentMetaData">
```

```
<xs:annotation>
383
384
                  <xs:documentation>
385
                      DocumentMetaData is the top-level element for the
386
     hData meta data specification. It is
387
                      embedded with every Atom 1.0 Content node.
388
                  </xs:documentation>
389
              </xs:annotation>
390
              <xs:sequence>
391
                  <xs:element minOccurs="0" name="PedigreeInfo" type="hd-</pre>
392
     md:PedigreeInfo">
393
                      <xs:annotation>
394
                           <xs:documentation>
395
                               This optional node holds the pedigree
     information for the Section Document.
396
397
                           </xs:documentation>
398
                      </xs:annotation>
399
                  </xs:element>
400
                  <xs:element name="DocumentId" type="xs:string">
401
                      <xs:annotation>
402
                           <xs:documentation>
403
                               This required element holds an identifier for
404
     the Section Document. It MUST be unique over any given
405
                               Section feed.
406
                           </xs:documentation>
407
                      </xs:annotation>
408
                  </xs:element>
409
                  <xs:element minOccurs="0" name="LinkedDocuments">
410
                      <xs:annotation>
411
                           <xs:documentation>
                               This optional node holds a list of URI links
412
413
     to documents that are related to this
414
                               Section Document. Use depends on the
     semantics of the Section Document Type.
415
416
                           </xs:documentation>
417
                      </xs:annotation>
418
                      <xs:complexType>
419
                           <xs:sequence>
420
                               <xs:element maxOccurs="unbounded" name="Link"</pre>
421
     type="hd-md:LinkInfo"/>
422
                           </xs:sequence>
423
                      </xs:complexType>
424
                  </xs:element>
425
                  <xs:element name="RecordDate">
                      <xs:annotation>
426
427
                           <xs:documentation>
428
                               This required node holds the information
429
     about Document creation and modification.
430
                           </xs:documentation>
431
                      </xs:annotation>
432
                      <xs:complexType>
433
                           <xs:sequence>
434
                               <xs:element name="CreatedDateTime"</pre>
435
     type="xs:dateTime">
436
                                   <xs:annotation>
437
                                        <xs:documentation>
```

```
438
                                            This required element contains
     the dateTime of creation of this documment.
439
440
                                            If this document is not derived
441
     (see PedigreeInfo), this is the time of the
                                            creation of the original. If this
442
443
     document is derived from another origin, this element
444
                                            contains the date of derivation.
445
                                        </xs:documentation>
                                    </xs:annotation>
446
447
                               </xs:element>
                               <xs:element minOccurs="0" name="Modified">
448
449
                                    <xs:annotation>
450
                                        <xs:documentation>
                                            This optional node is first
451
     created when the document is changed for the first time.
452
                                            It contains a collection of
453
454
     modification dates with optional pedigree information of the
                                            modifier.
455
456
                                        </xs:documentation>
457
                                   </xs:annotation>
458
                                    <xs:complexType>
459
                                        <xs:sequence minOccurs="1"</pre>
460
     maxOccurs="unbounded">
461
                                            <xs:element</pre>
462
     name="ModifiedDateTime" type="xs:dateTime">
                                                 <xs:annotation>
463
                                                     <xs:documentation>
464
465
                                                         This required element
     record a dateTime when the document was modified.
466
467
                                                     </xs:documentation>
468
                                                 </xs:annotation>
                                            </xs:element>
469
                                            <xs:element minOccurs="0"</pre>
470
471
     name="PedigreeInfo"
472
                                                 type="hd-md:PedigreeInfo">
                                                 <xs:annotation>
473
474
                                                     <xs:documentation>
475
                                                         This optional node
     contains the pedigree information of the modifier.
476
477
                                                     </xs:documentation>
                                                 </xs:annotation>
478
479
                                            </xs:element>
480
                                        </xs:sequence>
481
                                    </xs:complexType>
482
                               </xs:element>
483
                           </xs:sequence>
484
                      </xs:complexType>
485
                  </xs:element>
486
                  <xs:element minOccurs="0" name="Confidentiality"</pre>
487
     type="xs:string">
488
                      <xs:annotation>
489
                           <xs:documentation>
490
                               This element contains controls for
491
     confidentiality - details are TBD.
492
                           </xs:documentation>
```

```
493
                      </xs:annotation>
494
                  </xs:element>
495
                  <xs:element minOccurs="0" name="AccessControl">
496
                      <xs:annotation>
497
                           <xs:documentation>
498
                               This element contains controls for access
499
     control - details are TBD.
500
                           </xs:documentation>
501
                      </xs:annotation>
502
503
                  </xs:element>
                  <xs:element minOccurs="0" name="Consent">
504
505
                      <xs:annotation>
506
                           <xs:documentation>
                               This element contains controls for consent -
507
     details are TBD.
508
509
                           </xs:documentation>
510
                      </xs:annotation>
511
512
                  </xs:element>
513
              </xs:sequence>
514
              <xs:attribute name="MediaType" type="xs:string">
515
                  <xs:annotation>
516
                      <xs:documentation>
517
                          This attribute contains the media type of the
518
     document itself. If it is not present, the
519
                          default media type of the content type is
520
     assumed.
521
                      </xs:documentation>
522
                  </xs:annotation>
523
              </xs:attribute>
524
              <xs:attribute name="ContentType" type="xs:anyURI"</pre>
525
     use="optional">
526
                  <xs:annotation>
527
                      <xs:documentation>
528
                           This attribute contains the URI for the content
529
     type of this document. If it is not present,
530
                           the content type for the Section is implied. Note
     that the current hData Content Profiles assume
531
532
                           that the content type for all Section Documents
533
     within a given Section is uniform.
534
                      </xs:documentation>
535
                  </xs:annotation>
536
              </xs:attribute>
537
         </xs:complexType>
538
         <xs:complexType name="PedigreeInfo">
539
              <xs:annotation>
540
                  <xs:documentation>
541
                      This node contains the pedigree information.
542
                  </xs:documentation>
              </xs:annotation>
543
544
              <xs:sequence>
545
                  <xs:element minOccurs="0" name="XmlSignature"</pre>
546
     maxOccurs="unbounded">
547
                      <xs:annotation>
```

```
548
                          <xs:documentation> This optional node contains
549
     the signature information on
550
                               the document or this meta data.
551
     </xs:documentation>
552
                      </xs:annotation>
553
                      <xs:complexType>
554
                          <xs:sequence>
                               <xs:element name="Signature"</pre>
555
556
     type="ds:Signature">
557
                                   <xs:annotation>
                                       <xs:documentation> This Signature
558
     MUST contain: 1. a valid Reference
559
560
                                            to either the metadata or the
     Section Document 2. the ds:KeyInfo
561
                                            for the signer (optional with
562
563
     DSig - required here)
564
                                       </xs:documentation>
565
                                   </xs:annotation>
566
                               </xs:element>
567
                          </xs:sequence>
568
569
570
                          <xs:attribute name="documentMethod">
571
                               <xs:annotation>
572
                                   <xs:documentation>This optional attribute
573
     indicates what method was used
574
                                   to transform binary Section Document
575
     mediatypes into XML files for
576
                                   signature. Currently the only permitted
577
     methods are xml and base64.
578
                                   xml is the default XML signature over XML
     documents. base64 encodes
579
                                   a data stream into an XML document. The
580
581
     root node it root and
                                   contains the BASE64 encoded data.
582
     </xs:documentation>
583
584
                               </xs:annotation>
585
                               <xs:simpleType>
586
                                   <xs:restriction base="xs:string">
                                       <xs:enumeration value="base64"/>
587
                                       <xs:enumeration value="xml"/>
588
589
                                       <xs:enumeration value="sha256"/>
590
                                   </xs:restriction>
591
                               </xs:simpleType>
592
                          </xs:attribute>
                      </xs:complexType>
593
594
                  </xs:element>
595
                  <xs:element minOccurs="0" maxOccurs="1" name="Source">
596
                      <xs:annotation>
597
                          <xs:documentation>This node indicates the source
     of this data. </xs:documentation>
598
599
                      </xs:annotation>
600
                      <xs:complexType>
601
                          <xs:sequence>
```

```
602
                               <xs:element name="PedigreeInfo" type="hd-</pre>
     md:PedigreeInfo" minOccurs="0"/>
603
604
                               <xs:element maxOccurs="unbounded"</pre>
605
     minOccurs="0" name="Document"
606
                                   type="hd-md:LinkInfo"/>
607
                           </xs:sequence>
                           <xs:attribute name="derived" type="xs:boolean">
608
                               <xs:annotation>
609
                                   <xs:documentation>If the data is derived
610
611
     (i.e. copied or compiled from other sources) this attribute MUST be
     set to true. </xs:documentation>
612
613
                               </xs:annotation>
614
                           </xs:attribute>
                      </xs:complexType>
615
                  </xs:element>
616
617
                  <xs:element minOccurs="0" name="Author" type="xs:string">
618
                      <xs:annotation>
                           <xs:documentation>The identifier of the creators
619
620
     of this document. For derived documents, this is the author. Note
621
     that this identifier can identify machines as well as humans.
622
     </xs:documentation>
623
                      </xs:annotation>
624
                  </xs:element>
625
                  <xs:element minOccurs="0" name="Organization"</pre>
626
     type="xs:string">
                      <xs:annotation>
627
628
                          <xs:documentation>This element identifies the
629
     organization. </xs:documentation>
630
                      </xs:annotation>
631
                  </xs:element>
632
             </xs:sequence>
         </xs:complexType>
633
634
         <xs:complexType name="LinkInfo">
635
             <xs:sequence>
636
                  <xs:element name="Target" type="xs:anyURI"/>
                  <xs:any max0ccurs="unbounded" min0ccurs="0"/>
637
638
              </xs:sequence>
639
         </xs:complexType>
     </xs:schema>
640
```

4.3 Common Data Types

641 642

This section contains the normative schema for the common data types (see Section 3).

```
643
     <?xml version="1.0" encoding="UTF-8"?>
644
     <!-- Copyright 2009 The MITRE Corporation
645
646
     Licensed under the Apache License, Version 2.0 (the "License");
     you may not use this file except in compliance with the License.
647
648
     You may obtain a copy of the License at
649
650
    http://www.apache.org/licenses/LICENSE-2.0
651
652
     Unless required by applicable law or agreed to in writing, software
653
    distributed under the License is distributed on an "AS IS" BASIS,
```

```
654
     WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
655
     implied.
656
     See the License for the specific language governing permissions and
657
     limitations under the License. -->
658
     <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"</pre>
659
     elementFormDefault="qualified"
660
     targetNamespace="http://projecthdata.org/hdata/schemas/2009/06/core"
       xmlns:core="http://projecthdata.org/hdata/schemas/2009/06/core">
661
       <xs:element name="date" type="xs:dateTime"/>
662
663
       <xs:element name="dateRange">
664
         <xs:complexType>
665
           <xs:attribute name="high" type="xs:dateTime"/>
666
           <xs:attribute name="low" use="required" type="xs:dateTime"/>
667
         </xs:complexType>
668
       </xs:element>
669
       <xs:complexType name="codedValue" >
670
        <xs:simpleContent >
671
         <xs:extension base="xs:string">
672
           <xs:attribute name="code" />
           <xs:attribute name="codeSystem" />
673
674
           <xs:attribute name="version" ></xs:attribute>
675
           <xs:attribute name="displayName" />
676
         </xs:extension>
677
         </xs:simpleContent>
678
       </xs:complexType>
679
       <xs:element name="name">
680
         <xs:complexType>
681
           <xs:sequence>
682
             <xs:element name="title" type="xs:string"/>
683
             <xs:element name="given" type="xs:string"</pre>
684
     maxOccurs="unbounded"/>
             <xs:element name="lastname" type="xs:string"/>
685
686
             <xs:element name="suffix" type="xs:string"/>
687
           </xs:sequence>
688
         </xs:complexType>
689
       </xs:element>
690
       <xs:element name="address">
691
692
         <xs:complexType>
693
           <xs:sequence>
              <xs:element name="streetAddress" minOccurs="0"</pre>
694
695
     maxOccurs="unbounded" type="xs:string"/>
696
              <xs:element name="city" type="xs:string"/>
697
              <xs:element minOccurs="0" name="stateOrProvince"</pre>
698
     type="xs:string"/>
             <xs:element name="zip" minOccurs="0" type="xs:string"/>
699
700
              <xs:element minOccurs="0" name="country" type="xs:string"/>
701
           </xs:sequence>
702
         </xs:complexType>
703
       </xs:element>
704
       <xs:element name="telecom">
705
706
         <xs:complexType>
707
           <xs:attribute name="value" use="required"/>
708
           <xs:attribute name="use" use="required"/>
```

```
709
         </xs:complexType>
710
       </xs:element>
711
       <xs:element name="actor" abstract="true">
712
         <xs:annotation>
713
           <xs:documentation>
714
                  An actor is a generic type used to define various
715
     entities within the document. This will generally be a person, such
716
     as a
717
                  point of contact, doctor, gaurdian ..., or an
718
     organization, such as insurance provider, care provider ...
719
              </xs:documentation>
720
         </xs:annotation>
721
       </xs:element>
722
       <xs:complexType name="actor">
723
         <xs:sequence>
            <xs:element ref="core:actor"/>
724
725
         </xs:sequence>
726
       </xs:complexType>
727
       <xs:element name="organization" substitutionGroup="core:actor"</pre>
728
     type="core:organization.class"/>
729
       <xs:element name="person" substitutionGroup="core:actor"</pre>
730
     type="core:person.class"/>
731
       <xs:complexType name="person.class">
732
         <xs:annotation>
733
           <xs:documentation>
734
                  Generic definition of a person. name address, contact
     information such as telephone , email \dots
735
736
                  This is represented as a class of object so the same
737
     basic person structure can be used to define
738
                  multiple types of elements , such as patient, author,
739
     . . . .
740
                  name: the name of the individual, see the definition of
741
742
     name
743
                  address: 0 or more address entries, see the definition of
744
     address
745
                  telecom: 0 or more telecom entries, see the definition of
746
     telecom
747
              </xs:documentation>
         </xs:annotation>
748
749
         <xs:sequence>
750
           <xs:element ref="core:name"/>
751
           <xs:element minOccurs="0" maxOccurs="unbounded"</pre>
752
     ref="core:address"/>
753
           <xs:element minOccurs="0" maxOccurs="unbounded"</pre>
754
     ref="core:telecom"/>
755
         </xs:sequence>
756
       </xs:complexType>
757
       <xs:complexType name="organization.class">
758
         <xs:annotation>
759
           <xs:documentation> Base class definition of an organization.
760
761
                  An organization can have 0 or more points of contact
762
     which are represented as poc elements of type person.class
763
```

```
764
                  name: the Name of the organization , simple String value
765
                  pointsOfContact: wrapper element around 0 or more poc
766
                 poc elements are structured as person.class elements
     elements.
767
                  address: 0 or more address elements for the given
768
     organization
769
770
              </xs:documentation>
771
         </xs:annotation>
772
         <xs:sequence>
           <xs:element name="name" type="xs:string"/>
773
           <xs:element name="pointOfContacts">
774
775
              <xs:complexType>
776
                <xs:sequence>
777
                  <xs:element minOccurs="0" maxOccurs="unbounded"</pre>
     name="pointOfContact" form="unqualified" type="core:person.class"/>
778
779
                </xs:sequence>
780
              </xs:complexType>
781
           </xs:element>
782
           <xs:element minOccurs="0" maxOccurs="unbounded"</pre>
783
     ref="core:address"/>
784
         </xs:sequence>
785
       </xs:complexType>
786
       <xs:element name="informationSource">
787
         <xs:complexType>
788
           <xs:sequence>
789
              <xs:element name="author" type="core:person.class"/>
790
              <xs:element ref="core:date"/>
              <xs:element name="reference" type="xs:string"/>
791
              <xs:element name="informant" type="core:actor"/>
792
793
           </xs:sequence>
794
         </xs:complexType>
       </xs:element>
795
796
797
       <xs:element name="description">
798
         <xs:complexType>
799
           <xs:sequence>
800
              <xs:element name="text" type="xs:string"/>
              <xs:element name="codedValue" minOccurs="0"</pre>
801
802
     maxOccurs="unbounded" type="core:codedValue"/>
803
           </xs:sequence>
804
         </xs:complexType>
805
       </xs:element>
806
807
808
809
       <xs:group name="abstractSection">
810
         <xs:annotation>
811
           <xs:documentation>
812
                This definition is used to add data structures that will be
813
     common accross all hData modules. An hData mudule implementation
                will simply need to add this definition as a reference to
814
     obtain all of the common behavior
815
              </xs:documentation>
816
817
         </xs:annotation>
818
         <xs:sequence>
```

5 Bibliography

[1] G. Beuchelt, R. Dingwell, A. Gregorowicz, and H. Sleeper, "hData Packaging and Network Transport Specification," The MITRE Corporation, 2009.