hData Record Format v0.9

Gerald Beuchelt, Robert Dingwell, Andrew Gregorowicz, Harry Sleeper

The MITRE Corporation
202 Burlington Rd.
Bedford, MA 01730
U.S.A.

1 Introduction

The hData Record Format (HRF) describes the XML representation of the continuity of care information in an electronic health record (EHR). The HRF is implemented through a component-specific XML documents that are linked and organized through a "master document". For better organization, the individual XML documents are put into a hierarchy, with the master document at the root of this hierarchy. While the HRF defines a core set of components, it is fully extensible and can easily be adopted for more complex situations.

This specification only describes the organization of data within an abstract hData Record (HDR). Another specification describes how a HDR is serialized [1].

1.1 Namespaces

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 semantically significant.

Namespace Prefix	Namespace URI	Description
hrf	http://projecthdata.org/hdata/schemas/2009/06/core	Namespace for elements in this document
hcp	http://projecthdata.org/hdata/schemas/2010/04/hcp	Namespace for hData Content Profile Description language
hrf-md	http://projecthdata.org/hdata/schemas/2009/11/metadata	Namespace for meta data
xs	http://www.w3.org/2001/XMLSchema	XML Schema namespace

1.2 Glossary (Non-Normative)

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

hData Record (HDR) - an single instantiation of the HRF.

hData Restful API (HRA) - the part of the hData specification that defines the basic HTTP-based API for accessing or modifying an HDR.

hData Specification - a normative specification that defines the HRF, the HRA, and a file-based serialization format.

hData Content Profile (HCP) - a profile of the medical content of an HDR. An HCP is specified separately from the HRF. The hData Project defines an initial HCP (iHCP) that covers the 35 data elements for EHRs/EMRs defined by the National Quality Foundation.

Electronic Medical Record (EMR) - the medical record or records of a single patient in the IT system of an actor (health provider, government entity, payer, etc.). In this definition, an HDR is a type of EMR.

Electronic Health Record (EHR) - the collection of all EMRs of a single patient, across organizational and national boundaries.

EHR System - An IT system that creates, stores, and manages EMRs.

Clinical Document Architecture (CDA) - an XML specification by Health Layer 7 (HL7) that is intended to be used for EMRs.

Continuity of Care Record (CCR) - a specification by ASTM that is intended to be used for summary/continuity of care documentation. A CCM is a type of EMR.

Continuity of Care Document (CCD) - a profile of the CDA that accommodates the medical information of the CCR.

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 HITSP/C80 and HITSP/C83.

MITRE/L32 (L32) - a significantly constrained profile of the C32 specification. L32 comes with a normative schema and can be mapped onto the HRF.

1.3 Notational Conventions

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

When describing concrete XML schemas, this specification uses the following notation: each member of an 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 – any schema fragment or other schema description is informational only.

2 Hierarchical Organization

The basic approach of the hData Record Format is to represent the medical data through linked XML documents, which are organized through an abstract hierarchy. The hData storage and network protocols map this abstract hierarchy to a concrete implementation, such as a directory folder or web resource hierarchy.

In order to be able to accommodate more complex situations, HRF was designed with a number of extension points that allow the definition and 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.

2.1 Overall Structure

At the root of the hierarchy is the root document (RD) and additional documents that have relevance to 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 within the hierarchy. Implementers can either extend existing component sections or define new sections. Such newly created sections MUST be registered in the RD to be accessible.

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 XML Schema.

2.2 Root Document

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 text element contains a unique identifier for the extension. It is RECOMMENDED to use an URI. For elements of content type “application/xml”, it is RECOMMENDED that the text element contains a URI that resolves to an XML Schema for instance documents in this extension.

- /hrf:extensions/hrf:extension/@contentType (OPTIONAL) - This attribute contains the content type that for all documents in a section that register with this extension. If the attribute is not present, the documents in the section MUST be of content type “application/xml”.
- /hrf:sections - This node contains references to all component-specific sections (hrf:section)
- /hrf:sections/hrf:section (OPTIONAL) - A hrf:section describes an abstract collection of data elements within an hData record.
- /hrf:sections/hrf:section/@path - This text attribute is path segment, used to construct the full path to the section from the root. Valid characters are [a-z][A-Z][0-9]. The full path to a section is obtained by starting with a forward slash (“/”), and concatenating the path segments, separated by forward slashes.
- /htf:sections/hrf:section/@extensionId - This identifier MUST be equal to the identifier of any of the registered extension elements. It describes the default contentType and hdataType for documents contained in this section. Note that the metadata for each individual document MAY override the default contentType or hdataType .
- /hrf:sections/hrf:section/@name (OPTIONAL) - Used for a human-friendly name to this section.
- /hrf:sections/hrf:section/@requirement (OPTIONAL) – this attribute indicates if a given section is required or optional. Valid values are “required” or “optional”. If this attribute is not present, the section is required. NOTE: This attribute is ignored in the root document, but only used for the hData Content Profile Description Language (see section 2.5).

The root document schema MAY be extended to support additional features such as e.g. a mechanism to record versions of the data contained in the document.

2.3 Sections

Section within a hData record form an abstract hierarchy, similar to the file folder structure commonly used in 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.

2.4 Section Documents

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 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.

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 XML fragment starting with <hrf-md:DocumentMetaData>

- /DocumentMetaData - DocumentMetaData is the top-level element for the hData meta data specification.
- /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 text element 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/RecordDate/CreatedDateTime - This required element of type <xs:dateTime> contains the dateTime of creation of this documment. 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/RecordDate/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/RecordDate/Modified/ModfiedDateTime - This required element of type <xs:dateTime> records a dateTime when the document was modified.
- /DocumentMetaData/RecordDate/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.

This is the schema for <hrf-md:LinkInfo>:

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

2.5 hData Content Profiles

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

To describe hData Content Profiles, the following schema is used for the HCP definition file:

- /hrf:hcp – the root element for a HCP definition file.
- /hrf:hcp/@name – a simple display name
- /hrf:hcp/@id – a URI identifying the hData Content Profile. It is RECOMMENDED to use a URL that can be resolved into the HCP definition document.
- /hrf:hcp/hrf:extensions – this element describes the extensions used in this HCP. It uses the same syntax as in the root document as described in section 2.2.
- /hrf:hcp/hrf:sections – this element describes the sections that are to be included in a hData record that claims conformance to the HCP. It uses the same syntax as in the root document as described in section 2.2. NOTE: the requirements attribute is being used in the HCP, as described above.

3 Common Data Types

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 schema.

3.1 Name

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

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.

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.

3.4 Person

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.

3.5 Actor

The Actor substitutionGroup is used to represent situations when an entity may be either a person or an organization.

3.6 Organiation

This element is used to represent an organization and the basic deogrphic information associated with the 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

3.7 CodedValue

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 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

3.8 Date

This element is used to represent a singular point in time.

- /hrf:date/text() – the value of the date in question in xsd:date format

3.9 DateRange

This element is used to represent a date range.

- /hrf:dateRange/hrf:low – the low end of the date range represented as an hrf:date element
- /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

3.10 InformationSource

This element is used to represent where the information in a section may have originated from.

- /hrf:/informationSource/hrf:author – (Optional) the author of the referenced document represented as an hrf:person element
- /hrf:informationSource/date - (Optional) the date the referenced document was created represented as an hrf:date element
- /hrf:informationSource/reference – (Optional) a reference to the document from which the section information was derived
- /hrf:informationSource/informant - the individual or organization who added the information to the record, represented as an hrf:actor as described above

3.11 Description

This element is used to represent a general purpose description element that can also contain coded information.

- /hrf:description/hrf:text – free text block
- /hrf:description/hrf:codedValue – list of codedValues that pertain to the description

3.12 AbstractSection

This abstract complex type is used to represent a set of common feature that all section documents should contain. As this is a complexType the xpath statements below are not accurate as hrf:abstractSection would need to be replaced with an actual instantiation of the complexType.

- /hrf:abstractSection/hrf:description – (Optional) as described above
- /hrf:abstractSection/hrf:informationSource – (Optional) as described above

4 Appendix A: Normative Schemas

4.1 Root Document

This section contains the normative schema for the root document (see Section 2.2).

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Copyright 2009 The MITRE Corporation

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied.
See the License for the specific language governing permissions and
limitations under the License. -->

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified"
targetNamespace="http://projecthdata.org/hdata/schemas/2009/06/core"
xmlns:core="http://projecthdata.org/hdata/schemas/2009/06/core">
  <xs:element name="root">
    <xs:complexType>
      <xs:all>
        <xs:element ref="core:id"/>
        <xs:element ref="core:version"/>
        <xs:element ref="core:created"/>
        <xs:element ref="core:lastModified"/>
```

```
        <xs:element ref="core:extensions"/>
        <xs:element ref="core:sections"/>
      </xs:all>
    </xs:complexType>
  </xs:element>
  <xs:element name="id" type="xs:string"/>
  <xs:element name="version" type="xs:string"/>
  <xs:element name="created" type="xs:date"/>
  <xs:element name="lastModified" type="xs:date"/>
  <xs:element name="extensions">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded"
ref="core:extension"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="extension">
    <xs:complexType mixed="true">
      <xs:attributeGroup ref="core:extension"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="sections">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded"
ref="core:section"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:attributeGroup name="extension">
    <xs:attribute name="contentType" type=”xs:string”
use=”optional”/>
</xs:attributeGroup>
  <xs:element name="section">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded"
ref="core:section"/>
      </xs:sequence>
      <xs:attribute name="path" use="required"/>
      <xs:attribute name="name" use="optional"/>
      <xs:attribute name="extensionId" use="required"/>
      <xs:attribute name=”requirement” use=”optional”>
        <xs:simpleType>
          <xs:restriction base="xs:token">
            <xs:enumeration value="mandatory"/>
            <xs:enumeration value="optional"/>
          </xs:restriction>
        </xs:simpleType>
```

```
      </xs:attribute>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

4.2 hData Content Profile Definition

This section contains the normative profile for the hData Content Profile defintion (see section 2.5).

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Copyright 2009 The MITRE Corporation

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied.
See the License for the specific language governing permissions and
limitations under the License. -->

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified"
targetNamespace="http://projecthdata.org/hdata/schemas/2010/04/hcp"
xmlns:hcp="http://projecthdata.org/hdata/schemas/2010/04/hcp"
xmlns:core="http://projecthdata.org/hdata/schemas/2009/06/core">
  <xs:element name="hcp">
    <xs:complexType>
      <xs:all>
        <xs:element ref="core:extensions"/>
        <xs:element ref="core:sections"/>
      </xs:all>
      <xs:attribute name="name" use="required" type="xs:string" />
      <xs:attribute name="id" use="required" type="xs:anyURI" />
    </xs:complexType>
  </xs:element>
</xs:schema>
```

4.3 Section Document Meta Data

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

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Copyright 2009 The MITRE Corporation

```

```
    Licensed under the Apache License, Version 2.0 (the "License");
    you may not use this file except in compliance with the License.
    You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

    Unless required by applicable law or agreed to in writing,
software
    distributed under the License is distributed on an "AS IS" BASIS,
    WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied.
    See the License for the specific language governing permissions
and
    limitations under the License. -->

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:hd-
md="http://projecthdata.org/hdata/schemas/2009/11/metadata"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"

targetNamespace="http://projecthdata.org/hdata/schemas/2009/11/metada
ta">
    <xs:import namespace="http://www.w3.org/2000/09/xmldsig#"
            schemaLocation="http://www.w3.org/TR/2002/REC-xmldsig-
core-20020212/xmldsig-core-schema.xsd"/>
    <xs:complexType name="DocumentMetaData">
        <xs:annotation>
            <xs:documentation>
                DocumentMetaData is the top-level element for the
hData meta data specification. It is
                embedded with every Atom 1.0 Content node.
            </xs:documentation>
        </xs:annotation>
        <xs:sequence>
            <xs:element minOccurs="0" name="PedigreeInfo" type="hd-
md:PedigreeInfo">
                <xs:annotation>
                    <xs:documentation>
                        This optional node holds the pedigree
information for the Section Document.
                    </xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="DocumentId" type="xs:string">
                <xs:annotation>
                    <xs:documentation>
                        This required element holds an identifier for
the Section Document. It MUST be unique over any given
                        Section feed.
                    </xs:documentation>
```

```
                </xs:annotation>
            </xs:element>
            <xs:element minOccurs="0" name="LinkedDocuments">
                <xs:annotation>
                    <xs:documentation>
                        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.
                    </xs:documentation>
                </xs:annotation>
                <xs:complexType>
                    <xs:sequence>
                        <xs:element maxOccurs="unbounded" name="Link"
type="hd-md:LinkInfo"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="RecordDate">
                <xs:annotation>
                    <xs:documentation>
                        This required node holds the information
about Document creation and modification.
                    </xs:documentation>
                </xs:annotation>
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="CreatedDateTime"
type="xs:dateTime">
                            <xs:annotation>
                                <xs:documentation>
                                    This required element contains
the dateTime of creation of this documment.
                                    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.
                                </xs:documentation>
                            </xs:annotation>
                        </xs:element>
                        <xs:element minOccurs="0" name="Modified">
                            <xs:annotation>
                                <xs:documentation>
                                    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.
                                </xs:documentation>
```

```
                            </xs:annotation>
                            <xs:complexType>
                                <xs:sequence minOccurs="1"
maxOccurs="unbounded">
                                    <xs:element
name="ModifiedDateTime" type="xs:dateTime">
                                        <xs:annotation>
                                            <xs:documentation>
                                                This required element
record a dateTime when the document was modified.
                                            </xs:documentation>
                                        </xs:annotation>
                                    </xs:element>
                                    <xs:element minOccurs="0"
name="PedigreeInfo"
                                        type="hd-md:PedigreeInfo">
                                        <xs:annotation>
                                            <xs:documentation>
                                                This optional node
contains the pedigree information of the modifier.
                                            </xs:documentation>
                                        </xs:annotation>
                                    </xs:element>
                                </xs:sequence>
                            </xs:complexType>
                        </xs:element>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element minOccurs="0" name="Confidentiality"
type="xs:string">
                <xs:annotation>
                    <xs:documentation>
                        This element contains controls for
confidentiality - details are TBD.
                    </xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element minOccurs="0" name="AccessControl">
                <xs:annotation>
                    <xs:documentation>
                        This element contains controls for access
control - details are TBD.
                    </xs:documentation>
                </xs:annotation>

            </xs:element>
            <xs:element minOccurs="0" name="Consent">
                <xs:annotation>
                    <xs:documentation>
```

```
                        This element contains controls for consent -
details are TBD.
                    </xs:documentation>
                </xs:annotation>

            </xs:element>
        </xs:sequence>
        <xs:attribute name="MediaType" type="xs:string">
            <xs:annotation>
                <xs:documentation>
                    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.
                </xs:documentation>
            </xs:annotation>
        </xs:attribute>
        <xs:attribute name="ContentType" type="xs:anyURI"
use="optional">
            <xs:annotation>
                <xs:documentation>
                    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.
                </xs:documentation>
            </xs:annotation>
        </xs:attribute>
    </xs:complexType>
    <xs:complexType name="PedigreeInfo">
        <xs:annotation>
            <xs:documentation>
                This node contains the pedigree information.
            </xs:documentation>
        </xs:annotation>
        <xs:sequence>
            <xs:element minOccurs="0" name="XmlSignature"
maxOccurs="unbounded">
                <xs:annotation>
                    <xs:documentation> This optional node contains
the signature information on
                        the document or this meta data.
</xs:documentation>
                </xs:annotation>
                <xs:complexType>
                    <xs:sequence>
                        <xs:element ref="ds:Signature">
                            <xs:annotation>
```

```
                                    <xs:documentation> 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)
                                    </xs:documentation>
                                </xs:annotation>
                            </xs:element>
                        </xs:sequence>


                        <xs:attribute name="documentMethod">
                            <xs:annotation>
                                <xs:documentation>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 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.
</xs:documentation>
                            </xs:annotation>
                            <xs:simpleType>
                                <xs:restriction base="xs:string">
                                    <xs:enumeration value="base64"/>
                                    <xs:enumeration value="xml"/>
                                    <xs:enumeration value="sha256"/>
                                </xs:restriction>
                            </xs:simpleType>
                        </xs:attribute>
                    </xs:complexType>
                </xs:element>
                <xs:element minOccurs="0" maxOccurs="1" name="Source">
                    <xs:annotation>
                        <xs:documentation>This node indicates the source
of this data. </xs:documentation>
                    </xs:annotation>
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="PedigreeInfo" type="hd-
md:PedigreeInfo" minOccurs="0"/>
                            <xs:element maxOccurs="unbounded"
minOccurs="0" name="Document"
                                type="hd-md:LinkInfo"/>
                        </xs:sequence>
```

```
                    <xs:attribute name="derived" type="xs:boolean">
                        <xs:annotation>
                            <xs:documentation>If the data is derived
(i.e. copied or compiled from other sources) this attribute MUST be
set to true. </xs:documentation>
                        </xs:annotation>
                    </xs:attribute>
                </xs:complexType>
            </xs:element>
            <xs:element minOccurs="0" name="Author" type="xs:string">
                <xs:annotation>
                    <xs:documentation>The identifier of the creators
of this document. For derived documents, this is the author. Note
that this identifier can identify machines as well as humans.
</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element minOccurs="0" name="Organization"
type="xs:string">
                <xs:annotation>
                    <xs:documentation>This element identifies the
organization. </xs:documentation>
                </xs:annotation>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="LinkInfo">
        <xs:sequence>
            <xs:element name="Target" type="xs:anyURI"/>
            <xs:any maxOccurs="unbounded" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```

5 Bibliography

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