

# hData Record Format v0.7

Gerald Beuchelt, Robert Dingwell, Andrew Gregorowicz, Harry Sleeper

The MITRE Corporation

202 Burlington Rd.

Bedford, MA 01730

U.S.A.

© 2009 The MITRE Corporation. All rights reserved.

## 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	<a href="http://projecthdata.org/hdata/schemas/2009/06/core">http://projecthdata.org/hdata/schemas/2009/06/core</a>	Namespace for elements in this document

### 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 Continuity of Care 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.

HRF only defines representations for the most common CCD constructs. In order to be able to accommodate more complex situations, it 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 master document (MD) 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 MD 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 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.
- /hrf:sections/hrf:section/@typeld - A unique identifier. This identifier MUST conform to the requirements for XML namespace identifiers.
- /hrf:sections/hrf:section/@name - Used for a human-friendly name to this section.

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.

## 2.3 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 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 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). All hData implementations MUST support the NQF-35 hData Content Profile.

# 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 attribute 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:telecom/@preferred – (OPTIONAL) Boolean attribute 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 Organisation

This element is used to represent an organization and the basic demographic 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:documentId"/>
        <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="documentId" 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>
```

```

254   <xs:element name="sections">
255     <xs:complexType>
256       <xs:sequence>
257         <xs:element minOccurs="0" maxOccurs="unbounded"
258 ref="core:section"/>
259       </xs:sequence>
260     </xs:complexType>
261   </xs:element>
262   <xs:attributeGroup name="extension">
263     <xs:attribute name="requirement" use="required">
264       <xs:simpleType>
265         <xs:restriction base="xs:token">
266           <xs:enumeration value="mandatory"/>
267           <xs:enumeration value="optional"/>
268         </xs:restriction>
269       </xs:simpleType>
270     </xs:attribute>
271   </xs:attributeGroup>
272   <xs:element name="section">
273     <xs:complexType>
274       <xs:sequence>
275         <xs:element minOccurs="0" maxOccurs="unbounded"
276 ref="core:section"/>
277       </xs:sequence>
278       <xs:attribute name="path" use="required"/>
279       <xs:attribute name="name" use="required"/>
280       <xs:attribute name="typeId" use="required"/>
281     </xs:complexType>
282   </xs:element>
283 </xs:schema>

```

## 4.2 Common Data Types

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

```

287 <?xml version="1.0" encoding="UTF-8"?>
288 <!-- Copyright 2009 The MITRE Corporation
289
290 Licensed under the Apache License, Version 2.0 (the "License");
291 you may not use this file except in compliance with the License.
292 You may obtain a copy of the License at
293
294 http://www.apache.org/licenses/LICENSE-2.0
295
296 Unless required by applicable law or agreed to in writing, software
297 distributed under the License is distributed on an "AS IS" BASIS,
298 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
299 implied.
300 See the License for the specific language governing permissions and
301 limitations under the License. -->
302 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
303 elementFormDefault="qualified"
304 targetNamespace="http://projecthdata.org/hdata/schemas/2009/06/core"
305 xmlns:core="http://projecthdata.org/hdata/schemas/2009/06/core">

```



```

306 <xs:element name="date" type="xs:dateTime"/>
307 <xs:element name="dateRange">
308   <xs:complexType>
309     <xs:attribute name="high" type="xs:dateTime"/>
310     <xs:attribute name="low" use="required" type="xs:dateTime"/>
311   </xs:complexType>
312 </xs:element>
313 <xs:complexType name="codedValue" >
314   <xs:simpleContent >
315     <xs:extension base="xs:string">
316       <xs:attribute name="code" />
317       <xs:attribute name="codeSystem" />
318       <xs:attribute name="version" /></xs:attribute>
319       <xs:attribute name="displayName" />
320     </xs:extension>
321   </xs:simpleContent>
322 </xs:complexType>
323 <xs:element name="name">
324   <xs:complexType>
325     <xs:sequence>
326       <xs:element name="title" type="xs:string"/>
327       <xs:element name="given" type="xs:string"
328 maxOccurs="unbounded"/>
329       <xs:element name="lastname" type="xs:string"/>
330       <xs:element name="suffix" type="xs:string"/>
331     </xs:sequence>
332   </xs:complexType>
333 </xs:element>
334
335 <xs:element name="address">
336   <xs:complexType>
337     <xs:sequence>
338       <xs:element name="streetAddress" minOccurs="0"
339 maxOccurs="unbounded" type="xs:string"/>
340       <xs:element name="city" type="xs:string"/>
341       <xs:element minOccurs="0" name="stateOrProvince"
342 type="xs:string"/>
343       <xs:element name="zip" minOccurs="0" type="xs:string"/>
344       <xs:element minOccurs="0" name="country" type="xs:string"/>
345     </xs:sequence>
346   </xs:complexType>
347 </xs:element>
348
349 <xs:element name="telecom">
350   <xs:complexType>
351     <xs:attribute name="value" use="required"/>
352     <xs:attribute name="use" use="required"/>
353   </xs:complexType>
354 </xs:element>
355 <xs:element name="actor" abstract="true">
356   <xs:annotation>
357     <xs:documentation>
358       An actor is a generic type used to define various
359 entities within the document. This will generally be a person, such
360 as a

```

```

361         point of contact, doctor, gaurdian ... , or an
362 organization, such as insurance provider, care provider ...
363     </xs:documentation>
364 </xs:annotation>
365 </xs:element>
366 <xs:complexType name="actor">
367     <xs:sequence>
368         <xs:element ref="core:actor"/>
369     </xs:sequence>
370 </xs:complexType>
371 <xs:element name="organization" substitutionGroup="core:actor"
372 type="core:organization.class"/>
373 <xs:element name="person" substitutionGroup="core:actor"
374 type="core:person.class"/>
375 <xs:complexType name="person.class">
376     <xs:annotation>
377         <xs:documentation>
378             Generic definition of a person. name address, contact
379 information such as telephone , email ...
380             This is represented as a class of object so the same
381 basic person structure can be used to define
382             multiple types of elements , such as patient, author,
383 ....
384
385             name: the name of the individual, see the definition of
386 name
387             address: 0 or more address entries, see the definition of
388 address
389             telecom: 0 or more telecom entries, see the definition of
390 telecom
391         </xs:documentation>
392     </xs:annotation>
393     <xs:sequence>
394         <xs:element ref="core:name"/>
395         <xs:element minOccurs="0" maxOccurs="unbounded"
396 ref="core:address"/>
397         <xs:element minOccurs="0" maxOccurs="unbounded"
398 ref="core:telecom"/>
399     </xs:sequence>
400 </xs:complexType>
401 <xs:complexType name="organization.class">
402     <xs:annotation>
403         <xs:documentation> Base class definition of an organization.
404
405             An organization can have 0 or more points of contact
406 which are represented as poc elements of type person.class
407
408             name: the Name of the organization , simple String value
409 pointsOfContact: wrapper element around 0 or more poc
410 elements. poc elements are structured as person.class elements
411             address: 0 or more address elements for the given
412 organization
413
414         </xs:documentation>
415     </xs:annotation>

```

```

416     <xs:sequence>
417       <xs:element name="name" type="xs:string"/>
418       <xs:element name="pointOfContacts">
419         <xs:complexType>
420           <xs:sequence>
421             <xs:element minOccurs="0" maxOccurs="unbounded"
422 name="pointOfContact" form="unqualified" type="core:person.class"/>
423           </xs:sequence>
424         </xs:complexType>
425       </xs:element>
426       <xs:element minOccurs="0" maxOccurs="unbounded"
427 ref="core:address"/>
428     </xs:sequence>
429   </xs:complexType>
430   <xs:element name="informationSource">
431     <xs:complexType>
432       <xs:sequence>
433         <xs:element name="author" type="core:person.class"/>
434         <xs:element ref="core:date"/>
435         <xs:element name="reference" type="xs:string"/>
436         <xs:element name="informant" type="core:actor"/>
437       </xs:sequence>
438     </xs:complexType>
439   </xs:element>
440
441   <xs:element name="description">
442     <xs:complexType>
443       <xs:sequence>
444         <xs:element name="text" type="xs:string"/>
445         <xs:element name="codedValue" minOccurs="0"
446 maxOccurs="unbounded" type="core:codedValue"/>
447       </xs:sequence>
448     </xs:complexType>
449   </xs:element>
450
451
452   <xs:group name="abstractSection">
453     <xs:annotation>
454       <xs:documentation>
455         This definition is used to add data structures that will be
456 common accross all hData modules.  An hData mudule implementation
457 will simply need to add this definition as a reference to
458 obtain all of the common behavior
459       </xs:documentation>
460     </xs:annotation>
461     <xs:sequence>
462       <xs:sequence minOccurs="0">
463         <xs:element ref="core:informationSource" minOccurs="0"/>
464         <xs:element ref="core:description" minOccurs="0"/>
465       </xs:sequence>
466     </xs:sequence>
467   </xs:group>
468 </xs:schema>
469

```

## 5 Bibliography

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