

Describing Electronic Health Records Using XML Schema

Bird LJ, Goodchild A and Sue H

CRC for Enterprise Distributed Systems (DSTC Pty Ltd)
Level 7, GP South, The University of Queensland, Qld, 4072, Australia.
Web: www.dstc.edu.au,
E-mail: {bird, andrewg, hoylen}@dstc.edu.au.

Abstract

The use of Electronic Health Records (EHRs) to store patient health information is becoming increasingly prevalent. In today's distributed healthcare environment, there are obvious benefits to be gained from being able to store, query and exchange this information between different health care sites. The information contained in these EHRs varies greatly in type - ranging from simple quantity values through to medical images. The clinical data models used to represent health data also varies greatly due to the different legislation, standards and health practices of different countries, states and health care facilities.

One approach to implementing EHRs is based on the Good Electronic Health Record (GEHR) architecture. This approach introduces the concept of 'clinical archetypes' to handle the variety in clinical data models. These clinical data models can be standardized independently of the EHR framework.

This paper describes the current Australian GEHR trial. In this trial, XML Schema is being used to represent the GEHR object model as well as the clinical archetypes. This paper describes how XML Schema is being used in the trial and the issues that are faced.

1. Introduction

One of the most significant activities of the healthcare industry is information management. An enormous amount of data is collected about patients for storage and analysis. For this reason, information technology is an extremely useful tool for the healthcare industry. However, there are many challenges which must be overcome before information technology solutions can be successfully deployed in the healthcare environment. These challenges relate to a wide variety of issues, including privacy, medico-legal requirements and the effect on existing clinical practices. With this in mind, this paper describes work done towards developing an Electronic Health Record (EHR) architecture. EHRs contain clinical information about patients and therefore form a critical part of a healthcare system.

An EHR system allows clinical information about patients to be stored and transmitted electronically. Currently, many doctors keep paper records about their patients. When clinical patient data is exchanged, for example between a hospital and a GP, it is usually done manually through the post, or at best by fax or phone. This makes it difficult to obtain accurate information in a timely fashion.

Most modern hospitals have computerised records. However, these systems are usually proprietary and often only serve one specific department within the hospital. Hospitals can have dozens of individual systems which do not interoperate with each other. A patient's health information profile can be spread out over a number of disparate systems, making it difficult for clinicians to capture a complete clinical history of a patient.

An interoperable electronic health record system has the potential to improve the provision of health care. It would allow clinicians to be able to access a more timely and complete picture of a patient's clinical history. Therefore, clinicians can make better informed healthcare decisions. Furthermore, as test results could be more easily shared between healthcare providers, costly duplication of tests and services could be avoided.

For interoperable electronic health records to become a reality, a common EHR standard is required. This paper describes a new approach to EHRs using the Good Electronic Health Record (GEHR) architecture. This approach makes use of XML Schema at two different levels. This paper introduces the GEHR framework in Section 2, and XML Schema in Section 3. We then describe how XML Schema is being used within the GEHR framework: at the level of the GEHR object model in section 4, and at the clinical archetype level in section 5. Concluding remarks are made in section 6.

2. GEHR

The Good Electronic Health Record (GEHR) is a metadata-based EHR framework. The framework is designed to be open and non-proprietary, which are important factors for promoting widespread adoption by users and vendors.

The current GEHR work is based on an earlier European EHR project. That project produced a comprehensive set of requirements as well as a framework. That project was also called GEHR, but its name stood for "Good European Health Record". The current Australian GEHR work continues the work of the European project, but has made significant improvements which makes the framework more flexible and implementable. In this paper, the term GEHR will refer to the Australian GEHR work unless otherwise stated.

The current GEHR work is being undertaken as a part of an Australian trial. This trial was commissioned by the Computing Group of the Royal Association of GPs in Australia (GPCG) and involves the collaboration of clinicians, systems architects and developers. The trial involves the development of an implementation of the GEHR model and integrating it into GP software packages from three independent vendors. The implementation of the GEHR model is being called the "GEHR kernel". The Titanium team at DSTC is a part of this collaborative effort, and is assisting with the XML related components.

There are many aspects to the GEHR framework. However, in this paper we will be concentrating on the application of XML Schema in the GEHR framework.

The most significant concept added to GEHR for the Australian trial is the concept of 'clinical archetypes'. Rather than prescribing a single and fixed clinical model, the GEHR Object Model (GOM) provides the "building blocks" by which any clinical model can be represented within the standardized framework. The clinical archetypes are then used to define how clinically valid structures are constructed out of the GOM primitives. This approach allows the process of standardizing the EHR framework to be separated from the process of standardizing clinical data models. The clinical data models may vary between different countries, states, and health care facilities (HCFs). This separation allows HCFs can store and exchange their clinical EHRs in an interoperable way, while still meeting local clinical data requirements.

One of the original requirements for archetypes was that they needed to be available as structured documents. This allows them to be authored and exchanged outside of the EHR system environment. While it is possible to develop a specialised GEHR archetype authoring language, a standard schema authoring language is more desirable as it is more likely that third party tools will support the language. We chose to use XML Schema as it is an industry standard for which many supporting tools are already appearing. Furthermore, by using XML Schema as a foundation, the exchange and validation of XML based EHR extracts will be facilitated in the future.

3. XML Schema

XML Schema is a language for describing the structure of XML documents[7]. XML Schema provides a rich set of data types, data constraints and data concepts for XML documents. XML Schema serves a similar purpose as the XML Document Type Definition (DTD) language, but is more richer and more powerful. Although DTDs will continue to exist, XML Schema should better meet the requirements of a wide-range of data-oriented applications that will use XML. Some of the advanced features that XML Schema provides are:

Rich Data Typing: XML Schema provides an extensive typing mechanism with an broad range of primitive types, such as numeric, date/time, binary, boolean, URIs. Furthermore, complex types can be built from the composition of other types (either primitive or complex). In particular, XML Schema uses a single inheritance model that allows type definitions to be created by restricting or extending other type definitions.

Support for Namespaces: XML Schema is namespace-aware, enabling elements with the same name to be used in different contexts. Additionally, schema types and elements can be included (or imported) from a separate XML schema using the same (or different) namespace.

Constraints: XML Schema provides an assortment of constraint types, including format-based ‘pattern’ constraints (e.g. “\d{3}[A-Z]{2}” represents three digits followed by two uppercase letters), key and uniqueness constraints, key references (foreign keys), enumerated types (value constraints), cardinality (or frequency) constraints and ‘nullability’.

Other Features: A number of other features, including anonymous type definitions, element content types, ‘Any’ elements and attributes, annotations, groupings and the use of derived types in instance documents, are also provided by XML Schema.

We are using XML Schema as a constraint language for GEHR. There are some limitations in trying to apply XML Schema in this manner, which will be described in the following sections. This is a unique application of XML Schema. XML Schema is designed to describe XML documents. However, in this application we are using XML Schema to model constraints in the GEHR model, which was not originally modelled as an XML document.

4. GEHR Object Model and XML Schema

The XML Schema is being used at two levels. Firstly for representing the constraints in the GEHR Object Model, as described in this section. Secondly, it is being used to represent the clinical archetypes, which will be described in section 5.

Central to the GEHR framework is the GEHR Object Model[4]. The GEHR Object Model (or GOM) defines the data structures representing the electronic health record. There are five levels of abstraction in the GOM: EHR, transaction, navigation, content, and data levels.

4.1 EHR Level

At the very top level is the EHR. The EHR is a container for a set of transactions. The GEHR framework does not mandate a single centralised EHR for patients. Multiple EHRs may exist for a patient and stored distributedly at a number of different health care facilities.

4.2 Transaction Level

The electronic health record consists of a collection of transactions. These transactions are typed. Persistent transactions contain information which remains valid in the long term, such as family history, chronic conditions etc. Event transactions are used for information whose validity is relatively

short-lived, such as the test results, a contact with a health care professional or a hospital admission. This is illustrated in figure 1.

Transactions are versioned. This is to satisfy medico-legal requirements. If a transaction is modified a permanent copy of the previous state of must be kept.

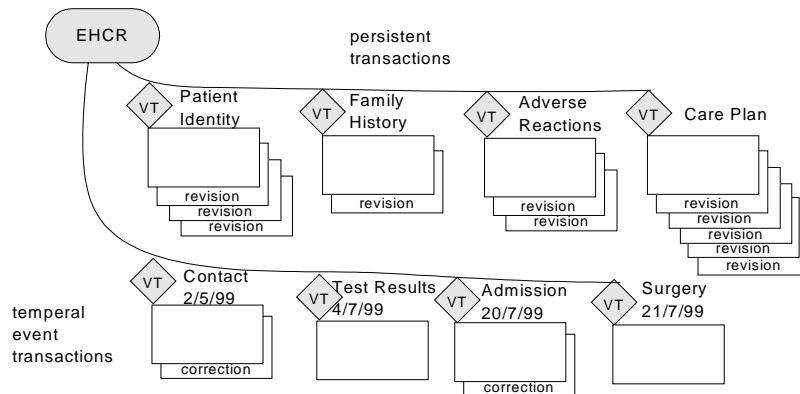


Figure 1 - Versioned transaction view of the EHR [based on figure 1 in (3)]

In XML Schema, the EHR can be modelled by creating a complex type. The versioned transactions, transactions and EHR content can be modelled using complex types. This is shown in the XML Schema fragment below:

```
<element name="GEHR" >
  <complexType>
    <element name="ehr_source" type="gom:G1_STRING"/>
    <element name="dt_creation" type="gom:G1_DATE_TIME"/>
    <element name="hcp_created_by" type="gom:G1_STRING"/>
    <element name="subject_transaction" type="gom:G1_VERSIONED_TRANSACTION"/>
    <element name="demographic_entities"
      type="gom:G1_VERSIONED_TRANSACTION_LIST" minOccurs="0"/>
    <element name="persistent_clinical_transactions"
      type="gom:G1_VERSIONED_TRANSACTION_LIST" minOccurs="0"/>
    <element name="event_clinical_transactions"
      type="gom:G1_VERSIONED_TRANSACTION_LIST" minOccurs="0"/> <!-- etc ... -->
  </complexType></element>

  <complexType name="G1_VERSIONED_TRANSACTION_LIST">
    <element name="transaction" type="gom:G1_VERSIONED_TRANSACTION" minOccurs="1"
      maxOccurs="unbounded"/>
  </complexType>

  <complexType name="G1_VERSIONED_TRANSACTION">
    <element name="gehr_version" type="gom:G1_STRING"/>
    <element name="time_created" type="gom:G1_DATE_TIME"/>
    <element name="access_rights" type="gom:G1_PERMISSIONS"/>
    <element name="amend_rights" type="gom:G1_PERMISSIONS"/>
    <element name="persistence_status" type="gom:G1_PERSISTENCE_STATUS" minOccurs="0"/>
    <element name="versions" type="gom:G1_VERSIONS"/>
  </complexType>

  <complexType name="G1_VERSIONS">
    <element name="transaction" type="gom:G1_TRANSACTION" maxOccurs="unbounded"/>
  </complexType>

  <complexType name="G1_TRANSACTION">
    <element name="locator_id" type="gom:G1_STRING"/>
    <element name="audit" type="gom:G1_COMMIT_AUDIT"/>
    <element name="content" type="gom:G1_EHR_CONTENT"/>
  </complexType>

```

```

</complexType>

<complexType name="G1_EHR_CONTENT">
  <group ref="gom:G1_ARCHETYPED"/>
  <element name="content" type="gom:G1_ORGANISER_ROOT"/>
  <element name="context" type="gom:G1_DEFINITION_CONTENT" minOccurs="0"/>
  <element name="is_persistent" type="gom:G1_BOOLEAN"/>
  <element name="persistence_status" type="gom:G1_PERSISTENCE_STATUS" minOccurs="0"/>
</complexType>

<group name="G1_ARCHETYPED">
  <element name="gehr_archetype_id" type="gom:G1_STRING"/>
  <element name="concept" type="gom:G1_PLAIN_TEXT"/>
</group>

```

4.3 Navigation Level

Each transaction consists of a collection of content items hierarchically grouped using *organisers*. Organisers are like headings in a paper-based report, which provide recognised structures. These structures are devised by clinical and medical practice bodies. For example, an organiser might consist of 'Problem' and SOAP ('Subjective', 'Objective', 'Assessment' and 'Plan') headings (as shown in figure 2).

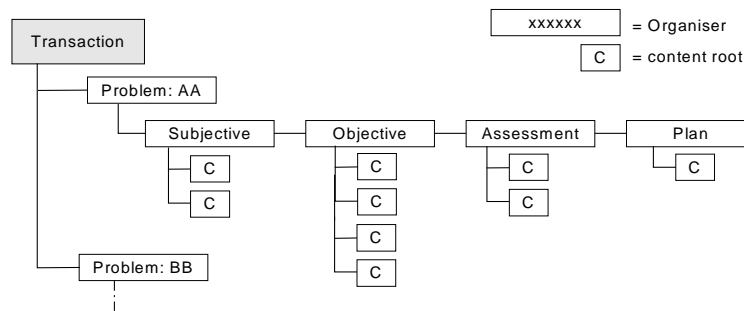


Figure 2 - Organiser view of a general practice contact transaction
[based on figure 8 from (3)]

In XML Schema, as illustrated below, this is modelled as complex types for organisers, organiser roots and organiser lists.

```

<complexType name="G1_ORGANISER">
  <element name="name" type="gom:G1_PLAIN_TEXT"/>
  <element name="content" type="gom:G1_DEFINITION_CONTENT_LIST"
    minOccurs="0"/>
  <element name="organisers" type="gom:G1_ORGANISER_LIST" minOccurs="0"/>
</complexType>

<complexType name="G1_ORGANISER_ROOT" base="gom:G1_ORGANISER"
  derivedBy="extension">
  <group ref="gom:G1_ARCHETYPED"/>
</complexType>

<complexType name="G1_ORGANISER_LIST">
  <element name="organiser" type="gom:G1_ORGANISER" maxOccurs="unbounded"/>
</complexType>

```

4.4 Content Level

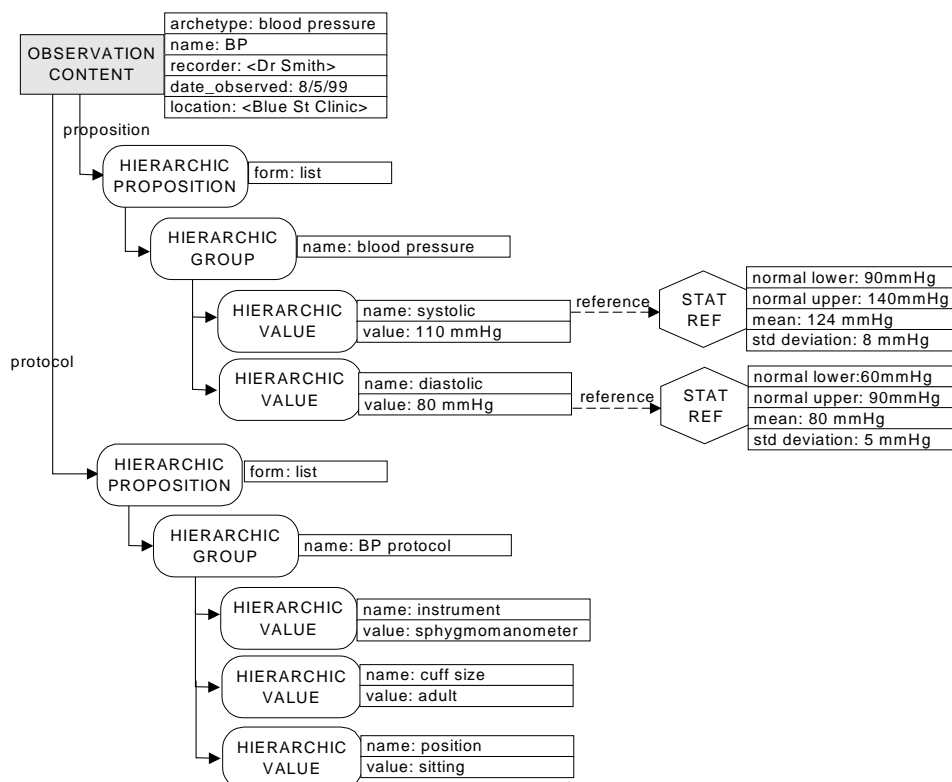


Figure 3 - An example of clinical content for blood pressure [based on figure 19 from (4)]

Clinical content expresses richly structured clinical data. These can be represented by various types of generic structures: including *proposition*, *observation*, *subjective*, *instruction* and *query*. These clinical structures are differentiated mainly by the contextual data they record. For example, observation items contain a timestamp and the identity of the recorder, while subjective items also record the identity of the information provider, and the degree of certainty that the clinician has the information correct. Figure 3 shows an example of clinical content, which represents a blood pressure observation.

At this level, the fine-grained record content is modelled to allow various clinical structures to be recorded. In its most general form, the structure is a *hierarchy*, but alternatives, such as *lists*, *time-series*, *tables* and *matrices* are also supported, allowing complex clinical data to be recorded in its most natural form. The final aspect of content structures is the provision of various ways of internal linking and cross-referencing items within a record, for example, to link events from different transactions such as an examination, test results and diagnosis into a single “problem” thread[2].

In XML Schema, as illustrated below, this is modelled by defining complex types for observation content, hierarchic proposition, hierarchic group and hierarchic value.

```
<complexType name="G1_DEFINITION_CONTENT">
  <group ref="gom:G1_ARCHETYPED"/>
  <element name="proposition" type="gom:G1_HIERARCHICAL_PROPOSITION"/>
</complexType>

<complexType name="G1_DEFINITION_CONTENT_LIST">
  <element name="content_item" type="gom:G1_DEFINITION_CONTENT"
    maxOccurs="unbounded"/>
</complexType>

<complexType name="G1_PREDICATE_CONTENT" base="gom:G1_DEFINITION_CONTENT"
  derivedBy="extension">
  <element name="context" type="gom:G1_ANY_CONTEXT"/>
  <element name="subject" type="gom:G1_STRING"/>
  <element name="protocol" type="gom:G1_HIERARCHICAL_PROPOSITION" minOccurs="0"/>
```

```

        maxOccurs="1"/>
</complexType>
<complexType name="G1_ANY_CONTEXT">
  <element name="recorder" type="gom:G1_STRING"/>
  <element name="comment" type="gom:G1_PLAIN_TEXT" minOccurs="0"/>
</complexType>
<complexType name="G1_OBSERVATION_CONTENT" base="gom:G1_PREDICATE_CONTENT"
  derivedBy="restriction">
  <element name="context" type="gom:G1_OBSERVATION_CONTEXT"/>
</complexType>
<!--etc ... -->
<complexType name="G1_HIERARCHICAL_PROPOSITION">
  <element name="form" type="gom:G1_PROPOSITION_FORMS" minOccurs="0"/>
  <element name="root" type="gom:G1_HIERARCHICAL_GROUP"/>
</complexType>
<complexType name="G1_HIERARCHICAL_ITEM" abstract="true">
  <element name="name" type="gom:G1_PLAIN_TEXT" minOccurs="0"/>
  <element name="context" type="gom:G1_ANY_CONTEXT" minOccurs="0"/>
  <element name="displayed" type="gom:G1_BOOLEAN" minOccurs="0"/>
</complexType>
<complexType name="G1_HIERARCHICAL_ITEM_LIST">
  <element name="hier_item" type="gom:G1_HIERARCHICAL_ITEM" maxOccurs="unbounded"/>
</complexType>
<complexType name="G1_HIERARCHICAL_GROUP"
  base="gom:G1_HIERARCHICAL_ITEM" derivedBy="extension">
  <element name="children" type="gom:G1_HIERARCHICAL_ITEM_LIST"/>
</complexType>
<complexType name="G1_HIERARCHICAL_VALUE" base="gom:G1_HIERARCHICAL_ITEM"
  derivedBy="extension">
  <element name="value" type="gom:G1_DATA_VALUE"/>
  <element name="reference" type="gom:G1_STATISTICAL_REFERENCE" minOccurs="0"/>
</complexType>

```

4.5 Data Level

The lowest level of the GOM defines the data types suitable for encoding the values of any clinical model, including plain text, term text, internal and external URIs, quantities, quantity ratios, quantity ranges, dates, times, durations, date/time ranges and multimedia[2].

The Data Level of the GOM uses the base XML Schema data types and from them constructs a number of EHR data types. For example, shown below are GEHR data values for quantities, quantity ratios, multimedia data, plain text and term text. These are all defined using the builtin XML Schema types of integer, float and string.

```

<complexType name="G1_DATA_VALUE"/>
<complexType name="G1_QUANTITY" base="gom:G1_DATA_VALUE"
  derivedBy="extension">
  <element name="value" type="gom:G1_REAL"/>
  <element name="precision" type="gom:G1_INTEGER" default="0" minOccurs="0"
    maxOccurs="1"/>
  <element name="units" type="gom:G1_STRING" minOccurs="0" maxOccurs="1"/>
</complexType>
<complexType name="G1_QUANTITY_RATIO" base="gom:G1_DATA_VALUE"
  derivedBy="extension">
  <element name="numerator" type="gom:G1_QUANTITY"/>
  <element name="denominator" type="gom:G1_QUANTITY"/>
</complexType>
<complexType name="G1_MULTIMEDIA_DATA" base="gom:G1_DATA_VALUE"
  derivedBy="extension">

```

```

<element name="external_reference" type="gom:G1_URI"/>
<element name="media_type" type="gom:G1_STRING"/>
<element name="size" type="gom:G1_INTEGER"/>
<element name="data" type="gom:G1_STRING" minOccurs="0" maxOccurs="1"/>
<element name="thumbnail" type="gom:G1_MULTIMEDIA_DATA"/>
</complexType>

<complexType name="G1_PLAIN_TEXT" base="gom:G1_DATA_VALUE"
  derivedBy="extension">
  <element name="value" type="gom:G1_STRING"/>
</complexType>

<complexType name="G1_TERM_TEXT" base="gom:G1_DATA_VALUE"
  derivedBy="extension">
  <choice>
    <group>
      <element name="value" type="gom:G1_STRING"/>
      <element name="primary" type="gom:G1_TERM_REF"/>
      <element name="qualifiers" type="gom:G1_TERM_REF_LIST" minOccurs="0"/>
    </group>
  </choice>
</complexType>

<complexType name="G1_TERM_REF">
  <element name="umls_entry" type="gom:G1_UMLS_ENTRY" minOccurs="0"/>
  <element name="code_in_term_set" type="gom:G1_STRING"/>
  <element name="term_set" type="gom:G1_TERM_SET_DESCRIPTOR"/>
  <element name="expansion" type="gom:G1_STRING"/>
</complexType>

<complexType name="G1_TERM_REF_LIST">
  <element name="term" type="gom:G1_TERM_REF" minOccurs="0"
    maxOccurs="unbounded"/>
</complexType>

<simpleType name="G1_INTEGER" base="integer"/>
<simpleType name="G1_REAL" base="float"/>
<simpleType name="G1_STRING" base="string"/>

```

4.6 Issues

In using XML Schema to represent the GOM, two main difficulties were identified. One relating to multiple inheritance and the other about uniqueness and foreign key constraints.

Firstly, the native GEHR Object Model uses multiple-inheritance to represent in some relationships. For example, a G1_ORGANISER_ROOT is both a G1_ORGANISER and a G1_ARCHETYPED. Because XML Schema uses single-inheritance it is not possible to model this subtype hierarchy in such a way that allows instances of G1_ORGANISER_ROOT to be replaced for either a G1_ORGANISER or a G1_ARCHETYPED.

Fortunately, for most subtypes in the GOM it was possible to choose one of the supertypes to inherit from and to include the elements from the other supertype in an element group. In this case, the subtype G1_ORGANISER_ROOT was made to be based on the supertype G1_ORGANISER, and then the other elements incorporated using the group ref=G1_ARCHETYPED. Effectively, this has removed the multiple-inheritance from the model.

The second aspect of the GOM that was difficult to represent in XML Schema were the unique and foreign key references. In XML Schema, foreign unique and foreign key references must be defined using XPath expressions to represent the element or attribute combinations that are either unique or key references. However, because the GEHR definition of G1_ORGANISER is recursive, there is no way of uniquely representing the element path to the elements inside a G1_DEFINITION_CONTENT. Consider for example how to represent the constraint that G1_PREDICATE_CONTENT's context/recorder must reference the key defined over:

"GEHR/demographic_entities/transaction/versions/transaction/ locator_id".

The foreign key field could be defined as:

GEHR/event_clinical_transactions/transactions/transaction/versions/transaction/content/content/ [{organisers/organiser/}] content/content_item/context/recorder.

However, as the element subpath "organisers/organiser" is optional and repeatable, this foreign key can not be represented as an XPath.

This illustrates that some forms of unique and foreign key constraints cannot be represented using XML Schema. Such constraints will have to be checked using another mechanism. For this project, these constraints will have to be checked in our software using specially written code.

Lastly, it should be noted that the data value G1_TERM_TEXT (which originally inherited from G1_PLAIN_TEXT in the GOM) was modelled in XML Schema using a "choice". The reason for this is was to enable the specification of valid term text enumerations in the archetypes. This will be explained in more detail in section 5.5 when we look at issues with archetypes.

5. GEHR Archetypes and XML Schema

5.1 Overview

It is impractical for the GOM to capture all the clinical concepts required for an EHR. While the GOM does express some clinical concepts (the idea of the record, transaction etc), it would be infeasible to include all currently used or potential clinical concepts in the model. A number of factors guarantee that it would be impossible to create a monolithic EHR standard:

There are an immense number of clinical concepts, corresponding to every kind of physical examination, pathology test result, anatomical terms, etc. The technical and political resources required to model the domain of clinical medicine would be enormous, and would be infeasible as a prerequisite step to establishing a health record standard.

- Mainstream technology changes quickly and can significantly affect procedures and methods (e.g. NMR, ultrasound, keyhole surgery, telemedicine), thus rendering existing clinical models obsolete or incomplete.
- Clinical norms change. For example, blood pressure was first recorded as phase I and phase IV and later as phase V. More recently the cuff size has to be recorded as well. Clearly it is inappropriate to constrain the recording of the measurement of blood pressure at the architectural level.
- Ongoing research is always contributing to better understanding of things, and hence changing the way medicine is done. For example, the change in the monitoring of diabetes from measuring urine glucose, to blood glucose and more recently to glycosylated haemoglobin.
- The social and cultural environment in which medicine is practised varies from country to country. Also procedures often change due to non-clinical factors, e.g. religious and cultural norms, influence of insurance companies, changes in federal and state health care policy etc.

Consequently, an EHR model which attempts to describe information structures for every possible clinical procedure or situation is fundamentally undesirable, since it would compromise widespread and future usability of the model. It would confuse the standardisation of the electronic health record with the standardisation of practice models, protocols, and other clinical concepts which are the business of clinical medicine at large[5].

In order to allow clinical structures to be implemented in the GOM, the concept of clinical *archetypes* was introduced. These clinical archetypes define the valid GOM information structures for particular clinical concepts such as "blood pressure", "family history" etc. Archetypes enable two

different processes of standardisation to occur independently - the technical process of standardisation of the GOM and related technologies, and the much broader process of clinical standardisation, which in a sense is never finished.

As an example, consider Figure 4, which illustrates the object structure of a blood pressure content item in a GEHR record. An archetype for “blood pressure with reference range and protocol” has been used to define the actual structure. In order to know how to construct a viable object model, the line between the model and meta-model must be explicitly defined; i.e. Which items should be modelled outside the GOM in archetypes, and which should remain inside? The answer to this question determines the limits of the concrete model. Based on experience in the GEHR project, it was found that archetypes are needed for:

- Transaction types, e.g. contact, summary.
- Navigational headings, e.g. organisers.
- Clinical content types e.g. lab-results, prescriptions, including their structure (list, table, series etc); this is by far the largest group;

These correspond to three of the levels of the GOM, previously described in section 4. Thus, in addition to the GOM definitions, we have a set of clinical archetype definitions. These archetypes are also expressed in XML Schema.

From the point of view of an EHR system, the archetypes are simply treated as input documents. The meta-model approach fulfills the requirement that GEHR does not constrain the way medicine is practised, but rather provides a way in which information created under any practice model can be stored[5].

The following sections illustrate how XML Schema can be used to define clinical archetypes for transactions, navigational headings and clinical content.

5.2 Transaction Archetypes

Transaction archetypes are clinical definitions which constrain the way in which the GOM’s transaction building blocks are constructed. For example, as illustrated below, a health organisation may define the notion of a “current medication” by restricting the transaction type to include medication specific organiser structures.

```
<complexType name="gehr.trans.persist.current_medication" base="gom:G1_TRANSACTION"
  derivedBy="restriction">
  <element name="content" type="cmed:Content"/>
</complexType>

<complexType name="Content" type="gom:G1_EHR_CONTENT" derivedBy="restriction">
  <element name="gehr_archetype_id" fix="gehr.trans.persist.current_medication"/>
  <element name="concept" type="cmed:ContConcept"/>
  <element name="content" type="cmed:ContCont"/>
  <element name="is_persistent" fix="TRUE"/>
</complexType>

<complexType name="ContConcept" base="gom:G1_PLAIN_TEXT" derivedBy="restriction">
  <element name="value" fixed="Current medication"/>
</complexType>

<complexType name="ContContent" base="gom:G1_ORGANISER_ROOT" derivedBy="restriction">
  <element name="gehr_archetype_id" fixed="gehr.org.current_medication"/>
</complexType>
```

5.3 Organiser Archetypes

The way in which navigational headings (ie organisers) are actually structured within a transaction, is defined within an archetype. For example, a current medication organiser structure, which includes both acute and ongoing medications could be defined as follows:

```
<complexType name="gehr.org.current_medication" base="gom:G1_ORGANISER_ROOT"
  derivedBy="restriction">
  <element name="name" type="med:medOrganiserRootName"/>
  <element name="content" maxOccurs="0"/>
  <element name="organisers" type="med:medOrganisers"/>
  <element name="gehr_archetype_id" fixed="gehr.org.current_medication"/>
  <element name="concept" type="med:medConcept"/>
</complexType>

<complexType name="medOrganiserRootName" type="gom:G1_PLAIN_TEXT"
  derivedBy="restriction">
  <element name="value" fixed="Current medication"/>
</complexType>

<complexType name="medOrganisers" type="gom:G1_ORGANISER_LIST" derivedBy="extension">
  <element name="ongoing" type="med:medOngoing"/>
  <element name="acute" type="med:medAcute"/>
</complexType>

<!-- Ongoing Medication Organiser -->
<complexType name="medOngoing" type="gom:G1_ORGANISER" derivedBy="restriction">
  <element name="name" type="med:medOngoingName"/>
  <element name="content" type="med:medOngoingContent"/>
  <element name="organisers" maxOccurs="0"/>
</complexType>

<complexType name="medOngoingName" type="gom:G1_PLAIN_TEXT" derivedBy="restriction">
  <element name="value" fixed="Ongoing medication"/>
</complexType>

<complexType name="medOngoingContent" type="gom:G1_DEFINITION_CONTENT_LIST"
  derivedBy="restriction">
  <element name="content_item" type="med:medOngoingContentItem"/>
</complexType>

<complexType name="medOngoingContentItem" base="gom:G1_DEFINITION_CONTENT"
  derivedBy="restriction">
  <element name="gehr_archetype_id" fixed="gehr.cont.instr.medication"/>
  <element name="concept" type="med:medOngoingContentItemConcept"/>
</complexType>

<complexType name="medOngoingContentItemConcept" type="gom:G1_PLAIN_TEXT"
  derivedBy="restriction">
  <element name="value" fixed="Ongoing medication"/>
</complexType>

<!-- Acute Medication Organiser -->
<complexType name="medAcute" type="gom:G1_ORGANISER" derivedBy="restriction">
  <element name="name" type="med:medAcuteName"/>
  <element name="content" type="med:medAcuteContent"/>
  <element name="organisers" maxOccurs="0"/>
</complexType>

<complexType name="medAcuteName" type="gom:G1_PLAIN_TEXT" derivedBy="restriction">
  <element name="value" fixed="Acute medication"/>
</complexType>

<complexType name="medAcuteContent" type="gom:G1_DEFINITION_CONTENT_LIST"
  derivedBy="restriction">
  <element name="content_item" type="med:medAcuteContentItem"/>
</complexType>

<complexType name="medAcuteContentItem" base="gom:G1_DEFINITION_CONTENT"
  derivedBy="restriction">
  <element name="gehr_archetype_id" fixed="gehr.cont.instr.medication"/>
```

```

    <element name="concept" type="med:medAcuteContentItemConcept"/>
  </complexType>

  <complexType name="medAcuteContentItemConcept" type="gom:G1_PLAIN_TEXT"
    derivedBy="restriction">
    <element name="value" fixed="Acute medication"/>
  </complexType>

  <!-- CONCEPT -->
  <complexType name="medConcept" base="gom:G1_PLAIN_TEXT" derivedBy="restriction">
    <element name="value" fixed="Current medication"/>
  </complexType>

```

5.4 Content Archetypes

The clinical content of a GEHR transaction is also modelled using archetypes. In the example below, we show a subset of the XML Schema for a blood pressure archetype. Blood pressure is modelled as a specialisation of the G1_OBSERVATION_CONTENT type.

```

<complexType name="gehr.cont.observe.bloodpressure" base="gom:G1_OBSERVATION_CONTENT"
  derivedBy="restriction">
  <element name="gehr_archetype_id" fixed="gehr.cont.observe.bloodpressure"/>
  <element name="concept" type="bp:bpConcept"/>
  <element name="context" type="gom:G1_OBSERVATION_CONTEXT"/>
  <element name="proposition" type="bp:bpProposition"/>
  <element name="subject" type="bp:bpSubject"/>
  <element name="protocol" type="bp:bpProtocol"/>
</complexType>

<!-- Concept . . . -->

<!-- Proposition -->
<complexType name="bpProposition" base="gom:G1_HIERARCHICAL_PROPOSITION"
  derivedBy="restriction">
  <element name="form" fixed="list"/>
  <element name="root" type="bp:bpPropositionRoot"/>
</complexType>

<complexType name="bpPropositionRoot" base="gom:G1_HIERARCHICAL_GROUP"
  derivedBy="restriction">
  <element name="name" type="bp:bpConcept"/>
  <element name="children" type="bp:bpPropositionChildren"/>
</complexType>

<complexType name="bpPropositionNoGenericChildren"
  base="gom:G1_HIERARCHICAL_ITEM_LIST" derivedBy="restriction">
  <element name="hier_item" minOccurs="0" maxOccurs="0"/>
</complexType>

<complexType name="bpChildren" base="bp:bpPropositionNoGenericChildren"
  derivedBy="extension">
  <element name="systolic" type="bp:systolicItem"/>
  <element name="diastolic" type="bp:diastolicItem"/>
</complexType>

<!-- Proposition: Systolic Item -->
<complexType name="systolicItem" base="gom:G1_HIERARCHICAL_VALUE"
  derivedBy="restriction">
  <element name="name">
    <complexType base="gom:G1_PLAIN_TEXT">
      <element name="value" fixed="systolic"/>
    </complexType>
  </element>
  <element name="value" type="bp:bpDataValue"/>
</complexType>

<!-- Proposition: Diastolic Item -->
<complexType name="diastolicItem" base="gom:G1_HIERARCHICAL_VALUE"
  derivedBy="restriction">

```

```

    <element name="name">
      <complexType base="gom:G1_PLAIN_TEXT">
        <element name="value" fixed="diastolic"/>
      </complexType></element>
    <element name="value" type="bp:bpDataValue"/>
  </complexType>

  <complexType name="bpDataValue" base="gom:G1_QUANTITY" derivedBy="restriction">
    <element name="value" type="bp:bpValue"/>
    <element name="units" type="unit:ISO_MMHG"/>
  </complexType>

  <simpleType name="bpValue" base="gom:G1_REAL">
    <minInclusive value="0"/>
    <maxInclusive value="1000"/>
  </simpleType>

  <!--Subject . . . -->
  <!--Protocol . . . -->
  <!-- . . . -->
  <complexType name="Position" base="gom:G1_HIERARCHICAL_VALUE" derivedBy="restriction">
    <element name="name">
      <complexType base="gom:G1_PLAIN_TEXT">
        <element name="value" fixed="position"/>
      </complexType></element>
    <element name="value" type="bp:PositionValue"/>
  </complexType>

  <complexType name="PositionValue" base="gom:G1_TERM_TEXT" derivedBy="extension">
    <choice>
      <group ref="bp:LyingTermText"/>
      <group ref="bp:StandingTermText"/>
      <group ref="bp:SittingTermText"/>
    </choice>
  </complexType>

  <group name="LyingTermText" >
    <element name="value" fixed="Lying">
      <element name="primary">
        <complexType base="gom:G1_TERM_REF" derivedBy="restriction">
          <element name="umls_entry" minOccurs="0" maxOccurs="0"/>
          <element name="code_in_term_set" fixed="75"/>
          <element name="term_set" type="GEHR-TermSet-v01"/>
          <element name="expansion" type="Lying down"/>
        </complexType>
      </element>
    </group>
  </complexType>

  <group name="StandingTermText" >
    <element name="value" fixed="Standing">
      <element name="primary">
        <complexType base="gom:G1_TERM_REF" derivedBy="restriction">
          <element name="umls_entry" minOccurs="0" maxOccurs="0"/>
          <element name="code_in_term_set" fixed="76"/>
          <element name="term_set" type="GEHR-TermSet-v01"/>
          <element name="expansion" type="Standing up"/>
        </complexType>
      </element>
    </group>
  </complexType>
  <!-- . . . -->

```

5.5 Issues

In designing the archetypes for the GPCG Trial, we encountered a number of concepts which were difficult to model in XML Schema. The approach which was taken was to follow the principal that the GOM definition was meant to provide the building blocks from which archetype definitions were to be

built. Therefore, we were trying to ensure that archetypes could be defined as “restrictions” on the complex types found in the GOM XML Schema. However, this proved to be difficult.

In some archetypes it was necessary to predefine the values that appeared in a list of items. For example, in the blood pressure archetype, the first value of the proposition root’s children should be fixed to “systolic” and the second value of this list should be fixed to “diastolic”. It was not possible to express this naturally, since it is not possible to repeat the same element in XML Schema. For example, the following approach is not valid:

```
<complexType name="bpPropositionChildren" base="gom:G1_HIERARCHICAL_ITEM_LIST"
derivedBy="restriction">
  <element name="hier_item" type="gom:SystolicItem"/>
  <element name="hier_item" type="gom:DiastolicItem"/>
</complexType>
```

The solution was to extend the hierarchical item list with the new values ‘SystolicItem’ and ‘DiastolicItem’ (as shown in Section 5.4). The same solution was also used to define a G1_ORGANISER_LIST (as shown in Section 5.3). The disadvantage of this approach is that we can not constrain extensions of a list to being of the type required by the GOM (in this case items of the list should be of type gom:G1_HIERARCHICAL_ITEM). This means that the GOM is no longer constraining the valid building blocks of the EHR, as originally intended.

A similar problem occurs when trying to specify some archetype constraints. In particular, we often need to restrict the value of a term text item to being from a set of valid coded term texts – for example, the value of “Patient Position” in the blood pressure’s protocol must be restricted to one of {(Lying, 75, GEHR-TermSet-v01), (Standing, 76, GEHR-TermSet-v01), (Sitting, 74, GEHR-TermSet-v01)}. While XML Schema allows enumeration constraints over simple types, there is no similar mechanism to define enumerations on complex types. Thus, it is not valid to write something like:

```
<complexType name="Position" base="gom:G1_HIERARCHICAL_VALUE" derivedBy="restriction">
  <element name="name">
    <complexType base="gom:G1_PLAIN_TEXT">
      <element name="value" fixed="position"/>
    </complexType>
  </element>
  <choice>
    <element name="value" type="bp:LyingTermText"/>
    <element name="value" type="bp:StandingTermText"/>
    <element name="value" type="bp:TermTermText"/>
  </choice>
</complexType>
```

Instead (as shown in Section 5.4), we specified valid sets of term texts by defining one ‘element group’ for each valid term text. These element groups contained the same sub-elements as a term text (i.e. value, primary, etc.), but had the values of these elements set appropriately for the valid term text being described. The disadvantage of this approach, once again, is that XML Schema is not used to restrict the archetype structures so that they conform to the types of the GOM.

Alternative methods were considered for representing these constraints (for example, equivalence classes and qualifying tags). However, all other methods which were considered either would not allow more than one term text constraint to be specified in a schema (in the case of equivalence classes), altered the structure of the instance documents from the original intent of the GOM, or offered no advantage over the approach chosen.

Thirdly, specifying key references from archetypes proved to be difficult. Consider, for example, trying to specify that the context/recorder of a blood pressure must reference the element “GEHR/demographic_entities/transaction/versions/transaction/locator_id”. Because it has been necessary to extend organiser definitions, rather than restrict them (to predefine lists of organisers), it is not possible for a content archetype to reference the full element path of the elements it contains. This means that it is not possible to specify key references inside content archetypes.

6. Conclusion

This paper has described how XML Schema has been used in GEHR. The GEHR framework with clinical archetypes is a practical approach to modelling electronic health records. Using XML Schema further enhances the ability to exchange and integrate EHRs, as well as leveraging off tools and expertise from the XML world.

This application of XML Schema is different from how XML Schema is normally used. Instead of describing an XML document, it is being used to describe a data model which is not XML based. This exercise has demonstrated that XML Schema has uses beyond just describing XML documents. We have found XML Schema to be a useful language for describing general data structures. However, there are some limitations in it which makes it difficult to represent certain constraints. Those limitations may also be a concern when using XML Schema for very complex XML documents.

The XML Schema specification is currently under development and is not yet finalised. While the current draft still has some limitations for non-traditional usage, XML Schema is proving to be a very useful language both in and beyond the XML world.

References

- 1 GEHR. (1999) GEHR Frequently Asked Questions:
URL: <http://www.gehr.org/introduction/faq-1.html>
- 2 Beale T. and Heard S. (1999) The GEHR Software Architecture.
URL: <http://www.gehr.org/technical/tb-tehre99-paper.html>
- 3 Beale T. and Heard S. (1999) The GEHR Technical Requirements:
URL: http://www.gehr.org/technical/requirements/gehr_requirements.html
- 4 Beale T. and Heard S. (1999) The GEHR Object Model Architecture.
URL: http://www.gehr.org/technical/model_architecture/gehr_architecture.html
- 5 Beale T. (1999) The GEHR Archetype System.
URL: http://www.gehr.org/technical/archetypes/gehr_archetypes.html
- 6 W3C. Extensible Markup Language (XML) 1.0.
URL: <http://www.w3.org/TR/1998/REC-xml-19980210>
- 7 W3C. XML Schema Part 0: Primer.
URL: <http://www.w3.org/TR/xmlschema-0/>

Acknowledgements

1. The authors would like to thank Thomas Beale (Deep Thought Informatics), Sam Heard (Ocean Informatics) and Zar Zar Tun (DSTC) for their contributions towards the work reported in this paper.
2. The work reported in this paper has been funded in part by the Cooperative Research Centres Program through the Department of the Prime Minister and Cabinet of the Commonwealth Government of Australia.