Solving the interoperability challenge of a distributed complex patient guidance system: A data integrator based on HL7’s Virtual Medical Record standard

Authors
Carlos Marcos\textsuperscript{1}, Arturo González-Ferrer\textsuperscript{2}, Mor Peleg\textsuperscript{2}, Carlos Cavero\textsuperscript{1}

Keywords
Electronic Health Records; Standards; Virtual Medical Record; Clinical Decision Support Systems; Interoperability

Word count (excluding title page, section titles, abstract, references, figures and tables, statements): 4278

Number of Figures: 5; Number of Tables: 4; Number of online supplements: 1

\textsuperscript{1} Postal address: C/ Albarracín, 25, 28037, Madrid, Spain
E-mail addresses: carlos.marcos@atos.net, carlos.cavero@atos.net
Department: Atos Research & Innovation
Telephone: +34 912148800
Institution: Atos Spain S.A.
City and Country: Madrid, Spain
* Corresponding author

\textsuperscript{2} Postal Address: Rabin Building, Room 7049, University of Haifa, Mount Carmel 3498838, Israel
E-mail addresses: arturogf@decsai.ugr.es, morpeleg@is.haifa.ac.il
Department: Information Systems
Institution: University Haifa
City and Country: Haifa, Israel
Abstract

Objective

We show how the HL7 virtual medical record (vMR) standard can be used to design and implement a data integrator (DI) component that collects patient information from heterogeneous sources and stores it into a personal health record (PHR), from which it can then retrieve data. Our working hypothesis is that the HL7 vMR standard in its release 1 version can properly capture the semantics needed to drive evidence-based clinical decision support systems (CDSS).

Materials and Methods

To achieve seamless communication between the PHR and heterogeneous data consumers, we used a three-pronged approach. First, the choice of the HL7 vMR as a message model for all components accompanied by the use of medical vocabularies eases their semantic interoperability. Second, the DI follows a service-oriented approach to provide access to system components. Third, an XML database provides the data layer.

Results

The DI supports requirements of a guideline-based CDSS implemented in two clinical domains and settings, ensuring reliable and secure access, high performance, and simplicity of integration, while complying with standards for the storage and processing of patient information needed for decision support and analytics. This was tested within the framework of a multinational project (www.mobiguide-project.eu) aimed at developing a ubiquitous patient guidance system (PGS).

Discussion
The vMR model with its extension mechanism is demonstrated to be effective for data integration and communication within a distributed PGS implemented for two clinical domains across different healthcare settings in two nations.

INTRODUCTION

Healthcare data is driving new generations of clinical decision support systems (CDSS) that are targeted not only at healthcare professionals but also at patients, with the aim of improving management of patients with chronic conditions as they go about their daily lives [1]. Such CDSS are based on electronic implementations of clinical practice guideline (CPGs), known as computer-interpretable guidelines (CIGs)[2]. Recently, a set of CDSS known as patient guidance systems (PGS) [3] have been developed for patients who require close monitoring. With PGS, patients carry small mobile sensors that can track the patient’s state and can react to situations that require the patient or a care provider to take action.

The data that serve as inputs to such PGS systems come from heterogeneous sources – not only hospital electronic medical records (EMRs), but also bio-sensor data, data entered by patients, and abstractions and recommendations inferred by the PGS regarding the patient. Indeed, with the right data, the systems can be used not only for personalized medicine, decision support, and better management of populations with chronic or complex conditions, but also secondary uses such as intelligent data analysis for compliance checking and for outcome research and quality improvement [4]. Yet the amount and heterogeneity of the data that must be integrated for such purposes, already daunting even for a single CIG and implementing organization, become immense when the system is scaled to several organizations and several CPGs. Consequently, data standards are required to simplify technical aspects of the system, including implementation, debugging, patient data access, and communication of recommendations from the PGS services [5], as well as its operation by the human stakeholders involved. Thus, novel data integration services, aligned with the selected standards, are required to
support the inter-communication of different components of the PGS system (e.g., care provider’s web interface, mobile phone patient interface, CDS engine).

We describe a data integrator (DI) component, based on the information model provided by the HL7 virtual medical record standard (vMR) [6], that can be used as gateway between data sources and PGS components, easing their interoperability. The DI encapsulates the data storage, hiding its complexity from the rest of the components, while at the same time providing an application programming interface (API) suitable for the implementation needs of the components that add new data (e.g., recommendations from the guideline-based CDS engine) and data consumers (a physician using a web portal to read those recommendations). Specifically, the DI provides service-oriented interfaces and transparent connection to a personal health record (PHR) where selected data can be stored. The DI has already been used as part of a real PGS (MobiGuide, described below) using the HL7 vMR standard [7] and its extension mechanisms, demonstrating that the complex semantics of a PGS system’s many interacting components can be captured.

**BACKGROUND AND SIGNIFICANCE**

**Patient Guidance Systems and PHRs: The example of MobiGuide**

PGSs are a modern form of CDSS that are targeted at patients as well as care providers. MobiGuide is a ubiquitous distributed CDSS [8] with over twenty components that are connected via service-oriented architecture (the project is funded by the European Union; www.mobiguide-project.eu). MobiGuide provides personalized clinical decision support that reacts to changes in the patient’s non-clinical context (e.g., travel may change the patient’s daily routine) and responds with appropriate recommendations [9]. Its data sources include hospital EMRs, processed bio-sensor data, patient-initiated input such as personal context changes and reporting of symptoms, and patient-specific
recommendations inferred by the CDSS component of the PGS and delivered to users (care providers or patients). In MobiGuide all of these data are integrated into a single PHR [10]. MobiGuide’s components must access the PHR in order to retrieve and insert patient clinical and demographic data provided by the DI, which serves as a single access point. The PHR component is also central in communicating events between system components and MobiGuide’s notification system.

![Figure 1. High-level architecture of a PGS (Patient Guidance System)](image)

Table 1 describes the main components of the MobiGuide system and how they interact with the DI. Figure 1 shows the MobiGuide’s high-level architecture.
<table>
<thead>
<tr>
<th>PGS component</th>
<th>Description</th>
<th>Interaction with DI</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data Integrator (DI)</strong></td>
<td>The DI components provide support for standardized data insertion/retrieval from the PHR. It also monitors data changes and notifies subscribed components when such changes occur. It imports and transforms data from hospital EMRs and stores the data in the PHR using the vMR format.</td>
<td>Not applicable</td>
</tr>
<tr>
<td><strong>Personal Health Record (PHR)</strong></td>
<td>A storage solution in XML format following the vMR schema with xQuery/xPath querying support. The DI is the only component allowed to access the PHR. It does this using xPath, xQuery, and xQuery Update.</td>
<td></td>
</tr>
<tr>
<td><strong>Backend DSS (DSS)/mobile DSS (mDSS)</strong></td>
<td>A distributed DSS engine (with both mobile and back-end server parts) which sends recommendations to patients and reacts to their inputs. Communicates with the patient via the PHR (making recommendations and reading the patient’s replies) and reacts to certain data patterns detected on the PHR.</td>
<td></td>
</tr>
<tr>
<td><strong>Body Area Network (BAN)</strong></td>
<td>A network of mobile bio-sensors and a smartphone application. Groups all requests from mobile components to access back-end components and routes them to the desired destination, like the DI, taking care of the technical issues involved (e.g., management of the online/offline status of the mobile device).</td>
<td></td>
</tr>
<tr>
<td><strong>Web Graphical User</strong></td>
<td>A web-based application used by the care provider to read data</td>
<td></td>
</tr>
<tr>
<td><strong>Interface (GUI)</strong></td>
<td>care provider to explore patient data and read/trigger notifications.</td>
<td>from the patient’s PHR and to read and trigger notifications sent by the DSS.</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------------------------------------------</td>
<td>-------------------------------------------------</td>
</tr>
<tr>
<td><strong>Mobile Graphical User Interface (GUI)</strong></td>
<td>The patient’s entry-point to the system, enabling him/her to insert and view data from the PHR.</td>
<td>Reads notifications and medical data from the PHR. It also inserts measurements and other related physiological data.</td>
</tr>
<tr>
<td><strong>QoD broker [11]</strong> (part of the mDSS)</td>
<td>Assesses the quality of the data inserted from the mobile phone and sensors so they can be used to impact clinical decision making.</td>
<td>Attaches QoD information to existing data on the PHR.</td>
</tr>
<tr>
<td><strong>Knowledge to Data Mapper (part of the knowledge)</strong></td>
<td>Closes the gap between the knowledge base handled by the DSS and the raw data in the PHR.</td>
<td>Transforms the knowledge-based requests coming from the DSS into queries expressed in term of raw data.</td>
</tr>
<tr>
<td><strong>Security components</strong></td>
<td>May provide single-sign-on capabilities, an audit trail, and other security-related services.</td>
<td>Stores the association between the patient identifier used within the EMR and the patient identifier used in the PHR and PGS, ensuring that if the PHR ID is compromised, the EMR data remains secure.</td>
</tr>
<tr>
<td><strong>Electronic Medical Record (EMR)</strong></td>
<td>The subsystem where the care provider’s institution stores its medical data.</td>
<td>The patient data needed for the PGS is copied from the EMR into the PHR at the moment of enrollment, and periodically refreshed to keep it up to date.</td>
</tr>
</tbody>
</table>

**Interoperability requirements of components**

In a PGS, component interoperability must be supported at both the syntactic and semantic levels [12].

Syntactic interoperability refers to low-level technical issues – for example, use of the same protocols...
and data formats. Semantic interoperability enables meaningful understanding by different systems of all information being transferred among components. Semantic interoperability is more difficult to implement, as it requires not only the definition of a common ontology or standard to be used by each of the services and components, but also its implementation separately by each one.

In previous work [3], we studied available standards for patient data models, including among others the HL7 vMR, openEHR archetypes [13], and ISO/CEN 13606 [14]. In that work we compared these standards in terms of expressiveness, user friendliness/ease of data representation, ability to link with EHRs CDSS, extendibility, provision of functionality for semantic integration, security and privacy, and scalability. We adopted the vMR as the logical data model for the entire MobiGuide system due to its simplicity as well as its expressiveness, which has been assessed as supporting representation of most of the data classes and their attributes used within the MobiGuide project. Appendix E presents our evaluation of the selected standard against criteria obtained from the Health IT Standards Committee (HITSC) [15] to address the HL7 vMR’s appropriateness for a pilot project like the one carried out in MobiGuide.

The HL7 Virtual Medical Record (vMR) Standard

The vMR standard was developed by HL7 [16] as a representation for data that is analyzed or produced by different CDS implementations (the present research uses release 1). The standard takes the rich semantic content of the HL7 version 3 reference information model (RIM) and expresses it in a format that hides the complexity of the full RIM and makes it easier to use by typical CDS knowledge engineers, reducing the likelihood of mistakes that may develop into potential patient safety hazards. The standard originated from academic research work [17] but has been extended based on a multi-institutional analysis of CDS data needs [6] encompassing twenty CDS systems [18]. The standard consists of a small set of classes, simplifying the learning curve for users and the time needed to represent different data
items. The model is composed of 22 classes and subclasses and is built upon two axes. The first represents the type of clinical information involved (eight high-level classes including Procedure, Observation, Problem, Substance Administration, AdverseEvent, Goal, Encounter, Supply), and the second the clinical workflow moment (e.g., Proposal, Order, Event), which implicitly represents the source of the information item (e.g., Proposal is produced by a system while Order is produced by a person).

Several papers have highlighted the potential role of a vMR as a solution to the diversity of terminology standards or CIG representations [19,20]; to link CDSS to clinical databases [21,22] or archetypes [3,23,24], overcoming the curly braces problem for institution-specific databases [25]; or even to extract quality indicators from EMRs [26]. Notably, the Health eDecisions (HeD) initiative [27] – a public-private initiative sponsored by the U.S. Office of the National Coordinator for Health IT (ONC) to develop scalable standards for sharing CDS – chose the vMR standard as its foundational data model. As part of this initiative, the vMR standard was enhanced in several ways [28]. Moreover, starting in 2014, ONC and the U.S. Centers for Medicare and Medicaid Services (CMS) have sponsored a follow-on initiative to HeD known as the Clinical Quality Framework (CQF), whose charge includes harmonizing standards for CDS and clinical quality measurement [29]. As a part of this initiative, the latest version of the vMR is being harmonized with the Quality Data Model (QDM) [30] to develop a successor to the vMR known as the HL7 Quality Improvement and Clinical Knowledge (QUICK) model [28].

Several recent projects have used the HL7 vMR as their foundational data model. Many of these employ OpenCDS (www.opencds.org), which provides a reference implementation of the vMR standard and is used by several organizations for clinical decision support and clinical quality measurement. For instance, the Immunization Calculation Engine (ICE) [31,32] uses OpenCDS to provide immunization CDS, including in a major commercial electronic health record system. To take another example, the Shared
Care Platform (SCP) [33] is a social network tool for physicians, who can use it to obtain and share clinical recommendations after mapping HL7 Continuity of Care Document (CCD) [34] data into a vMR format. Other research [35,36] has involved the mapping of existing patient data models into the HL7 vMR standard. Our study constitutes a step forward in standardizing the implementation of evidence-based CDSS and augments the variety of use cases for the HL7 vMR standard, validating its appropriateness in this scenario and identifying its benefits and limitations. In particular, our research involves modeling and aligning CIG knowledge with the standard data model, something not addressed in any related work.

We previously described our use of the HL7 vMR standard during the initial implementation of a scenario from a CPG for gestational diabetes mellitus (GDM) concerning detection and reporting of non-compliance with dietary recommendations [37]. In that research, we used Unified Modeling Language Sequence Diagrams [38] to show how the vMR model could be used as a mechanism for asynchronous communication among components of the MobiGuide system. While the analysis of the vMR included in this prior work [37] was based on initial implementation of one scenario of the GDM guideline, the current analysis is based on experiences gained through full implementation of the GDM guideline and initial implementation of an atrial fibrillation guideline. To our knowledge, this manuscript represents the most comprehensive analysis of the vMR to date in the context of clinical guideline representation and implementation.

**METHODS**

**Adding semantics to the vMR classes**

Although the HL7 vMR is an expressive data model designed to provide support for CDSS [6], we encountered clinical scenarios during our implementation of the MobiGuide PGS for which HL7 vMR
class attributes did not exist in the base model for the needed semantics. Some of these issues could be resolved by widening the semantics of existing attributes (e.g., the class observationOrder with a dataSourceType attribute having the value “DSS” can be used to indicate that the recommendation does not need a physician’s confirmation). In other cases, we rely on controlled vocabularies, such as SNOMED CT [39] and LOINC [40], to represent semantically meaningful data items, and use SNOMED’s post-coordination mechanism [41] to capture compositional semantics like the expression shown in the observationFocus attribute of Figure 2, expressing a measurement of blood glucose after dinner. Yet other scenarios required adding class attributes. In the HL7 vMR release 1, the RelatedClinicalStatement and RelatedEntity classes can be used to extend the model natively. However, the extension mechanism is verbose. The HL7 vMR release 2 incorporates a simpler extension mechanism that we ported into the release 1 to simplify our modeling. The added attributes are described in the Results section below. Technical details of this implementation are provided in Appendix A.
Figure 2. JSON representation of an ObservationProposal vMR instance originating from the DSS to carry out a post-prandial blood glucose measurement, including the post-coordinated SNOMED code for the observationFocus code. An extension is also included for the proprietary DssID that identifies the recommendation within the back-end DSS.
**Design principles for the DI**

The design principles for the DI address both syntactic and semantic interoperability, while recognizing that some of the MobiGuide components developed by third-party vendors use non-standard formats for storage and communication. First, by providing a **common point of connection** (API) to the components, it allows them to use any low-level protocol (e.g. REST Web Services [42]) to communicate, as long as they follow minimum requirements of standardization and security. Centralizing all communication through the DI has two main advantages: (i) the API provided by the DI can be reused later to integrate new incoming components; and (ii) improved privacy and security measurements can be easily implemented so patients can know who is accessing their data and why. This is a priority in the latest e-Health global directives [43].

Second, semantic interoperability is implemented by forcing the calling components to use a **unified message model**. The advantage of the DI in this respect is that iterative steps for specific protocol conversion can be built upon the existing and standardized APIs, allowing the developers of calling components to adapt to the message model step by step, from very specific and proprietary methods (e.g., insertFastingBloodGlucoseMeasurementResult) to more generic methods (e.g., insertObservation), up to a final generic implementation driven by the HL7 vMR standard patient information model (e.g., insertClinicalStatement).

Another design principle concerns the ability to export data from the various data sources to the PHR. Here, we followed a double approach. First, the standardization effort to homogenize how consumer components invoke the DI’s API resulted in a wrapper layer around the DI. Second, specific extractor classes were developed to convert the different hospital EMR data to vMR instances to ease their integration process.
Service-oriented data storage and retrieval

Figure 3 presents the internal architecture of the DI and its interfaces with other systems, while Table 2 lists the DI’s internal components and their functions. The PGS components send their data requests to the DI using the JavaScript Object Notation (JSON) [44] standard, which provides both light and robust communication over the Internet. The query results are returned as Java objects or as XML data containers ready to be inserted into the PHR back-end (vMR-Atoms). These Java objects are also automatically generated from the vMR schema using the Java built-in XML binding compiler (XJC)[45], resulting in a flexible and highly scalable system: any change or upgrade to the vMR schema is easily implemented on the wrapper layer.

Figure 3. DI structure and its interfaces with other systems
The DI’s extractor classes allow full retrieval from the EMR when a patient is enrolled into the system, and in addition support information refreshment of two types. For some components in the system, daily refresh may be sufficient (e.g., pattern recognition engines using historic data), while others may need the latest updated information on demand (e.g., the physician’s GUI).

Table 2: Functions of the DI’s internal components and of the PHR’s back-end

<table>
<thead>
<tr>
<th>DI sub-component</th>
<th>Description</th>
<th>Technical details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wrapper</td>
<td>Receives data requests</td>
<td>Built over an Axis2 web-service system [46], with the calls accepting HL7 vMR-compliant data fragments using JSON. To enable its use from the DI, we use the library GSON [47], which automatically converts the JSON-based requests encoded in Base64 into Java objects (see Figure of Appendix B). Besides the XML, each VMRAtom has a parameter detailing in XPath format [48] the precise point where that XML fragment should be inserted within the vMR tree as well as an update point. This enables the update of vMR items and consequently the refresh of information coming from the EMR. Further technical details are provided in Appendix B.</td>
</tr>
<tr>
<td>VMRAtoms: vMR-compliant XML data containers ready to be directly inserted into the PHR back-end.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Extractor</td>
<td>Retrieves data from already</td>
<td>Uses XML and HTTP protocols. Additional</td>
</tr>
</tbody>
</table>
established hospital EMRs into vMR-compliant format. technical details supplied in the figure of Appendix C.

| Notifier | This subsystem monitors incoming requests that modify PHR data in some way (insertion, update, or deletion), and notifies components of changes to data they have previously subscribed to. Before and after each modification of the data, the notifier is called from the main DI classes. Before the modification it stores the pre-modification status for comparison with the post-modification status. If there is any difference, subscribed components are notified. The notifier can also retrieve the new data and send it along with the notification. |
| PHR Back-end | An XML database which is straightforward to use because the vMR standard is provided as an XML Schema. Uses BaseX [49] – an open-source lightweight XML database which has evolved towards a database management system highly compliant with W3C directives, supporting the latest versions of XPath [48] and XQuery 3.0 [50] and including support for the XQuery Update 1.0 W3C recommendation. Further details are provided in Appendix D. |

For well-structured EMRs that are able to provide XML data resembling the vMR items, it is also possible to use DI functionality to directly transform the EMR data into the HL7 vMR structure. This is also useful for adapting old vMR-compatible EMRs to future releases. This transformation can be accomplished by
specifying the correspondence between EMR and vMR data in an XSLT file. Figure 4 and Figure 5 show a fragment of such a file.

![XSLT code example](image)

**Figure 4. Example of an EMR -> vMR transformation file, written in standard XSLT**

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet
 xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
 version="1.0" >
 <xsl:output
 method="xml"
 encoding="UTF-8" />
 <xsl:template match="/">
   <demographics>
     <xsl:apply-templates/>
   </demographics>
 </xsl:template>
 <xsl:template match="patient">
   <xsl:element name="part">
     <xsl:attribute name="datasource">EMR</xsl:attribute>
     <xsl:attribute name="type">GIV</xsl:attribute>
     <xsl:attribute name="value"></xsl:attribute>
     <xsl:value-of select="patient" />
   </xsl:element>
 </xsl:element>
</xsl:stylesheet>
```

**This XSLT will produce a vMR demographics tag**

**vMR structure with “name” and “part” subtags must be fulfilled**

**The EMR <patient> tag will be converted into a vMR <patient>**

![Diagram](image)

**Figure 5. Example of EMR -> vMR transformation**

For the current release of the DI, data from the hospital’s system is read-only, in keeping with the security policies of the participating healthcare institutions. In future releases a two-way connection (reading and insertion) is envisioned.
RESULTS

To validate the feasibility of the DI, we describe our experience with its implementation within MobiGuide.

MobiGuide is currently being tested among patients in Spain suffering from gestational diabetes mellitus and for patients in Italy with atrial fibrillation (AF). These domains pose different requirements: different hospitals, different clinical data, and different ways of representing and accessing this data. For example, for AF we had to represent substance administrations, while in the GDM domain we needed to represent different measurements made by the patient, such as blood glucose. Given the different requirements of the two domains and the low expertise in using the vMR standard among developers of external components, we recognized that it was important to implement the DI iteratively. At first, specific data access methods were implemented for each data item. This procedure helped developers of accessing components who were not used to the standard’s logical model. The final API uses a generic insertion and retrieval method that receives lightweight JSON formatted properly according to the standard and converted internally by the DI into vMR-conformant XML data. This greatly simplified integration, since the JSON chunks were modeled not by developers but by knowledge engineers while modeling the CIGs. To ensure the quality of this modeling, a specialist in finding vocabulary codes was tasked with providing these codes to a vMR encoder, and the semantics of the modeling were validated by the encoder along with all the involved developers. These information fragments were later used in the same way by different components, guaranteeing validity and coherence when writing into the PHR (e.g., using the same SNOMED/UMLS codes; using the same attributes for similar data items).

Moreover, the modeled JSON fragments and the lists of vocabulary codes were used when designing the conversion of hospital EMR data.
Since the preliminary study reported in [37], we have nearly completed the implementation of the GDM guideline and begun implementation of the AF guideline. Based on this work and that reported in [37], we can draw several conclusions about use of the HL7 vMR standard during implementation of the two CPGs.

First, we now have a clearer view as to which vMR classes support the distribution of decision support among the components of the system. Table 3 presents the classes and number of instances used for the GDM domain. Many of these classes are used for messaging between components, using the vMR axis of workflow-related information that implicitly represents the source of the data. This messaging support allows components to interconnect and enables to audit each step that takes place during the decision support workflow. For example, proposal-order-result is a common pattern of HL7 vMR data fragments that is repeated in many scenarios to communicate a recommendation of the DSS (e.g., changing frequency of blood glucose measurements due to good glycemic control), its confirmation by the physician, and finally a response from the patient.

<table>
<thead>
<tr>
<th>vMR Class</th>
<th>Num. instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObservationResult</td>
<td>34</td>
</tr>
<tr>
<td>ObservationProposal</td>
<td>13</td>
</tr>
<tr>
<td>ProcedureProposal</td>
<td>10</td>
</tr>
<tr>
<td>ProcedureOrder</td>
<td>9</td>
</tr>
<tr>
<td>ProcedureEvent</td>
<td>16</td>
</tr>
</tbody>
</table>
Second, we have identified the areas where using the standard’s extension mechanisms is needed to simplify workflow-related scenarios and to capture detailed semantics such as transaction time, quality of data, patient preferences, and the identifiers of clinical guideline steps from which recommendations originate. These areas are shown in Table 4, which extends partial information available in Table III of [37]. In this respect, the ongoing development of the HL7 vMR Templates [51] may simplify the extension mechanism. However, these templates were not available during the modeling of our vMR instances. For example, the template “DeliverableMessage” (category “communication”, templateID 2.16.840.1.113883.3.1829.11.12.2.3) could support message sending, while “PartOfAnEncounter” (category “Relationship”, templateID 2.16.840.1.113883.3.1829.11.16.2.2) could easily specify related statements (e.g. lab results) resulting from an encounter.

In addition to the extensions, we had to use proprietary code systems in order to identify patterns not captured either by standard terminologies or by post-coordination (e.g., “AF episode of 1-minute detected using the Bruce Protocol by the Linker algorithm”).

Table 4. Areas where the HL7 vMR extension mechanisms were used or where the model was updated in order to augment the semantics of existing vMR classes during implementation of the MobiGuide project.

<table>
<thead>
<tr>
<th>vMR Class Affected</th>
<th>Description</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Added CodedNameValuePair to vMR datatypes</td>
<td>Added to make the XML more compact, thus providing extension mechanism to ClinicalStatement and EntityBase classes, and to ensure future compatibility with the latest HL7 vMR release, which already supports an</td>
</tr>
</tbody>
</table>
“attribute” extension mechanism.

<table>
<thead>
<tr>
<th>EntityBase</th>
<th>Used extension mechanism with references to previously defined CodedNameValuePair</th>
<th>Enables inclusion of MobiGuide specific attributes like enrollmentDate, and qualityOfData attributes (e.g., “measurement accuracy”) [11].</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClinicalStatement</td>
<td>Added the TransactionTime attribute</td>
<td>Converts our vMR-based storage solution into a temporal database, thereby improving simplicity and performance (as this element must be accessed frequently by the processing subcomponents, extensions would otherwise be required by each relevant subclass).</td>
</tr>
<tr>
<td>ClinicalStatement</td>
<td>Used extension mechanism with references to previously defined CodedNameValuePair</td>
<td>Enables inclusion of PGS additional attributes like GuidelineID and GuidelineStepID (see also [24]).</td>
</tr>
<tr>
<td>ObservationOrder</td>
<td>Added the observationSig attribute</td>
<td>Enables representation of patient preferences, also stored in the PHR. Added directly to the schema instead of using RelatedClinicalStatement or RelatedEntity extensions for simplicity.</td>
</tr>
</tbody>
</table>

DISCUSSION

In this paper, we evaluate the use of the HL7 vMR standard for data integration and communication within a distributed multi-component PGS. Our system demonstrates that the standard facilitates interoperability across settings, organizations, and nations, and shows that the vMR data model can cope with the needs of a DI used to collect and distribute required information among system components. When used together with controlled terminologies, not only does the HL7 vMR standard provide an appropriate logical level for integration, but it also facilitates implementation of the software components and easy adaptation of existing APIs of those components, given the use of standards also at the lower levels (XML schema, XQuery, JSON, etc.). We also show in this paper how these standards can be combined to support primary use of healthcare data, with different types and modalities for different use cases, including personalized patient and care provider decision support for chronic and complex diseases. In light of the overlap in content across terminology and data model standards and multiple possible linkages between them, we propose a method of ordering these standards that should
be applied when the vMR is used to represent patient data, including guidance as to how the terminology should fit into the vMR information model [52] – from widening the meaning of existing vMR attributes, to the use of post-coordination to capture complex meanings of clinical statements, and finally to adding new attributes to vMR classes to add semantics that cannot otherwise be captured.

**Specific strengths and limitations**

Based on our experience implementing the vMR standard in MobiGuide, we have found evidence of the following strengths and limitations:

**Strengths**

- The vMR standard is expressive enough to represent a wide range of clinical and demographic information.

- It is possible to *automatically* map some EMR information into the HL7 vMR, provided that it is stored in a schema similar in structure. We successfully implemented XSLT transformations for such cases.

- Representing the HL7 vMR as JSON code permits simple and standardized insertion of data items into the PHR by the different MobiGuide components (which have different programmers and technologies). In all cases the programmers, who all lacked previous experience with the HL7 vMR, were able to adapt to it and develop the protocols on their components, thus supporting interoperability. We consider this proof of the usability of the standard for real use cases.

- Using the vMR data model, we were able to produce a lightweight communication system that is ultimately converted to XML data instances stored in a commercial XML database which can be queried using XQuery 3.0 and XPath standards.
To guarantee patients’ privacy and security, demographic data must be stored separately from clinical data. The vMR schema enables strict separation of such data in separate XML branches without significantly impacting any of the calling components.

Limitations

- Learning the semantics of the vMR standard is complicated. However, in our experience, only the knowledge engineers have to understand the complex semantics in order to generate JSON code that can be used by the developers of the DI and the calling components.

- The use of the vMR model necessitates not only the modeling of vMR instances but also the alignment of guideline authoring tools with the standard’s logical model. When the knowledge base specifies a clinical recommendation it must be aligned with the type of recommendation (e.g., ProcedureProposal, ObservationOrderProposal), and callback messages must be placed between CDSS components to make the DSS engine aware of special events (e.g., did the physician accept a recommendation previously sent by the DSS so that the guideline can continue to be interpreted?). In MobiGuide, the GESHER authoring tool [53] was extended in order to accommodate the vMR logical model.

- Specification of some items, like therapies, resulted in vMR structures that were too verbose, which could lead later to performance issues. The use of more refined structures (for example, the already mentioned use of extension mechanisms instead of creating large RelatedClinicalStatement/RelatedEntities structures) or faster databases may partially or completely solve such issues. The design of new classes as the standard evolves, like those in the Quality Improvement DAM [54] (e.g., StatementOfOccurrence plus statement modality and topic) may also simplify the modeling of periodic therapies (or any periodic recommendation).
• We found some performance limitations in the use of BaseX, especially when the system must cope with massive insertions of data. However, given the low number of patients simultaneously using the system and their rates of data insertion, this situation is only probable when simulation data are used during the testing phase. We plan to study these limitations in more detail and explore possible workarounds (like the use of another no-SQL database, a lightening of DI structure, or activation of a bulk insertions API).

• The use of the standard also raises some technical difficulties related to the need to create many vMR instances that must be modeled by knowledge engineers with development skills. For example, the GDM guideline needed modeling of 95 data instances (see Figure 2), as shown in Table 3. This is because many more data items need modeling when the standard is to be used as both a message model between components and as a storage model for the PHR. With regard to the technical difficulties of creating many data instances following the schema, we believe this problem relates less to the standard itself than to the low number of toolsets available to ease this modeling task (we had to use XML/JSON editors as Oxygen [55]).

In addition to these practical limitations of the vMR, our study also had some limitations:

• The HL7 vMR is still an evolving standard. Recent versions allow extensions more easily than release 1. However, adapting our system fully to the HL7 vMR release 2 was not trivial due to changes in the structure of classes and new attributes not considered in release 1.

• Our study examined two clinical domains and a single use (CDSS). Hence it is likely that our experience is not exhaustive. We are currently exploring use of the vMR model for secondary purposes, such as cohort identification [24], phenotype extraction [56,57], compliance checking [58], and CPG improvement based on mining the relationships between patient context, clinical actions taken, and outcome assessment [59].
Implications for ongoing standard development

Based on what we have learned, we would like to point out the following implications for ongoing standard development. First, simplicity of the standard is essential, both to enable users to learn it effectively and rapidly, and to ensure high performance in storing and retrieving data.

Second, different uses of the standard, such as multiple end-user roles, multiple sources of data, and different notions of time, require adding semantics to the existing vMR classes. Maintaining extension mechanisms in future vMR versions is critical. The ongoing development of the HL7 vMR Templates [51] could simplify the way in which extensions can be made. The use of templates may steepen the learning curve for users, but this should be offset by increased development speeds, reductions in the number of bugs during modeling, and potentially, easier data querying. Moreover, the knowledge acquisition process could be supported by development of an indexed portal of HL7-based templates, similar to those provided by openEHR [13] or the Clinical Information Modeling Initiative (CIMI) [60] through the Clinical Knowledge Manager (CKM) or the CIMI-Browser, respectively.

CONCLUSION

Based on development iterations for two clinical domains of a DI using the vMR standard as a logical and physical model within the MobiGuide project, we conclude that the use of a standardized message model is mandatory for ensuring interoperability in a PGS system that includes heterogeneous interacting components. The HL7 vMR standard with its extension mechanism has proven successful in this respect.
FUNDING STATEMENT

This work was supported by the European Union’s Seventh Framework Programme for research, technological development and demonstration grant number 287811.

COMPETING INTERESTS STATEMENT

We have no competing interests to declare.

CONTRIBUTORSHIP STATEMENT

Carlos Marcos led the work on the technical background of the DI and its subsystems (wrapper, extractor, notifier, and PHR) and wrote related parts of the paper, including the description of the interfaces with the rest of the components.

Arturo González-Ferrer handled most of the standardization procedures, worked side by side with developers from Atos during design of the DI functionalities, and wrote major portions of the Method and Results sections.

Mor Peleg wrote major parts of the Introduction, Discussion, and Background.

Carlos Cavero wrote those portions of the paper related to the storage solution (BaseX) used to implement the PHR and their interfaces with the DI (XPath/xQuery).

REFERENCES


44  Crockford D. The application/json media type for javascript object notation (json). Internet Eng.

45  ORACLE. Java Architecture for XML Binding Compiler.


51  HL7. HL7 Version 3 Standard: Clinical Decision Support; Virtual Medical Record (vMR) Templates,

52  Richesson RL, Krischer J. Data standards in clinical research: gaps, overlaps, challenges and future

and maintenance of procedural and declarative clinical decision-support knowledge. Open Med

http://www.hl7.org/special/Committees/projman/searchableProjectIndex.cfm?action=edit&Proj-
etNumber=1045


records for high-throughput phenotyping: the SHARPn consortium. J Am Med Informatics Assoc

57  Richesson RL, Hammond WE, Nahm M, et al. Electronic health records based phenotyping in
next-generation clinical trials: a perspective from the NIH Health Care Systems Collaboratory. J
Am Med Inf Assoc 2013;20:e226–e231.

58  Panzarasa S, Quaglini S, Sacchi L, et al. Data mining techniques for analyzing stroke care
2010. 939–43.

Appendix A. Technical details of implementation of vMR extensions

In cases where the vMR standard schema must be extended, the Java structure of classes containing the vMR subclasses must be recreated, which is done, externally to the Wrapper Layer by using the binding compiler tool (XJC)[1] of standard Java Architecture for XML Binding (JAXB)[2]. Once this is done, the resultant Java classes are ready to be fed onto the GSON library which instantiates and populate them according to the data received in JSON format. Later on, vMR-compliant XML can be extracted from the populated Java Objects by using JAXB and encapsulated into VMRAtoms.

Appendix B. Technical details of the Wrapper

Figure 1 shows how the incoming requests from the calling components are transformed into atomic objects, which can then be inserted into the PHR. These objects, named VMRAtom, are HL7 vMR-compliant XML data containers ready to be directly inserted into the PHR back-end.

The wrapper can be easily modified to accept as input also pure XML format, while the DI has already included a parameter to choose the format of the response (XML or JSON). The outcomes of the WrapperLayer are VMRAtoms.
Figure 6: Detail of the WrapperLayer parsing a JSON request into vMR-compatible Java objects
Appendix C. Technical details of the Extractor classes

Figure 2: Detail of one of the Extractor classes converting a JSON request into vMR-compatible Java objects

Appendix D. Technical Details of the XML-based PHR Back-end

The HL7 vMR standard is provided as XML Schema, which makes the use of XML databases in a straightforward way. BaseX [3] is an open-source light-weight XML database which has evolved towards a database management system highly compliant with W3C directives, supporting the latest versions of XPath [4] and XQuery 3.0 [5] and including also support for the XQuery Update 1.0 W3C recommendation. The database can be run as a stand-alone module on Windows or Linux machines or
deployed as a Web Application Archive (WAR) on most common web servers, like Apache Tomcat [6]. The built-in indexes can be used to speed-up complex XQueries and XPath expressions.

BaseX provides three different ways of accessing the data: REST interface, Java Client or Xquery for Java (XQJ), with the DI implementing all of them and using Java Client as the default. Although any of the explained choices of deployment can be used, the current implementation assumes that both the BaseX and the DI are deployed as WARs on the same web server, which in this case is a Tomcat v7.0 running over a Linux Virtual Machine.

**Appendix E. Evaluation of HL7 vMR against HITSC Standard Evaluation Criteria.**

At the August 30, 2012 meeting, the Health IT Standards Committee’s (HITSC) Nationwide Health Information Network Power Team (NwHIN) developed recommendations for standards evaluation criteria [7] and they were sent to the Office of the National Coordinator for Health IT (ONC).

These criteria aim to cover maturity and adoptability requirements. We express in parenthesis the consensus evaluation of the vMR standard that three of the co-authors of the paper (AGF, CM, MP) conducted. This evaluation considers the description of the metrics Low, Moderate, High, where descriptions of these metrics for each criterion are available in the Transmittal Letter of the HITSC NwHIN, from August 30, 2012 (http://www.healthit.gov/facas/health-it-standards-committee/health-it-standards-committee-recommendations-national-coordinator).

<table>
<thead>
<tr>
<th>- Maturity Criteria:</th>
</tr>
</thead>
<tbody>
<tr>
<td>o <em>Maturity of Specification</em></td>
</tr>
<tr>
<td>▪ Breadth of support (M)</td>
</tr>
<tr>
<td>▪ Stability (M)</td>
</tr>
<tr>
<td>▪ Adoption of Specification (M)</td>
</tr>
<tr>
<td>o <em>Maturity of Underlying Technology Components</em></td>
</tr>
<tr>
<td>▪ Breadth of Support (H)</td>
</tr>
<tr>
<td>▪ Stability (M-H)</td>
</tr>
<tr>
<td>▪ Adoption of Technology (M)</td>
</tr>
<tr>
<td>▪ Platform Support (H)</td>
</tr>
<tr>
<td>▪ Maturity of the Technology Within its Life Cycle (L-M)</td>
</tr>
</tbody>
</table>
- **Market Adoption**
  - Installed Health Care User Base (M)
  - Installed User Base Outside of Health Care (L)
  - Interoperable Implementations (M)
  - Future projections and anticipated support (H)
  - Investments in User Training (M)

- **Adoptability Criteria:**
  - **Ease of Implementation and Deployment**
    - Availability of off-the-shelf infrastructure to support implementation (H)
    - Standard as Success Factor (M)
    - Conformance criteria and tests (M-H)
    - Availability of reference implementations (M)
    - Quality and Clarity of Specifications (M-H)
    - Specification Modularity (M-H)
    - Separation of Concerns (H)
    - Ease of Use of Specification (M)
    - Degree to which Specification uses familiar terms to describe “real-world” concepts (H)
    - Runtime Decoupling (H)
    - Appropriate Optionality (L-M)
  - **Ease of Operations**
    - Comparison of targeted scale of deployment to actual scale deployed (L)
    - Number of operational issues identified in deployment (M)
    - Degree of peer-coordination needed (M)
    - Operational scalability (i.e., operational impact of adding a single node) (L-M)
    - Fit to Purpose (M)
  - **Intellectual Property**
    - Openness (H)
    - Affordability (H)
    - Licensing Permissiveness (M)
    - Copyrights Centralization (H)
    - Freedom from Patent Impediments (H)

Classification:
According to the evaluation we did and the classification proposed in the NwHIN transmittal letter into three possible stages (emerging standard, pilots, national standard), the standard is suitable for being used for pilots, which is the final aim of MobiGuide. In several metrics (as degree of optionality or fit to purpose) we think that newer releases of the standard or the new harmonization effort with quality models (QIDAM, QUICK) [8] will improve the path to be used as national standard.

Other criteria

There are other criteria that have been used as requirement to technical standards alike, as those enumerated in the section 3 of the QIDAM standard [8] submitted for ballot in August 2014. Many of them overlap with those expressed by the HITSC. We include some of these metrics, those that we considered appropriate for evaluation not fully covered by HITSC, valued in a similar fashion as before (from low to high):

- Appropriate Coverage (H)

- Suitable for Extension/Refinement (H)
- Supports mechanism for defining templates (H)

- Represents the canonical basis of clinical concepts (H)

Format (H):

- VMR Release 1 is both a logical and physical model expressed in XML, being able to be use directly for a pilot.

Usability (M-H):

- Intuitive structure and naming of classes and attributes (M-H)
- Effectiveness. Ensure that the standards allow users to achieve their goals accurately (M-H)
- Efficiency. Ensure that this will be done in an efficient manner (overlaps to some extent with the intuitive structure, suitability for extension and “degree of optionality” in the HITSC criteria). (M)

Computability (H):

- Semantic clarity (M-H). Represent clinical concepts and attributes in an unambiguous manner.
- “Just enough” concept granularity (H).
- Inferencing (H). The model defines concept relationships (e.g. is-a and part-of) needed by CDS use cases.
- Performance (M). Although this metric is directly related with the concrete technology used for implementing the standard (which is, of course, not part of the standard itself), we have found some cases where the model structure causes the resulting document to be excessively verbose, which can lead to poor performance on the transmission and processing of it. The flexibility provided by the extension mechanism, though, can be used to partially limit this.

REFERENCES


8 HL7. HL7 Domain Analysis Model: Health Quality Improvement, Release 1. 2014.