

# Use of the Virtual Medical Record Data Model for Communication among Components of a Distributed Decision-support System

Arturo González-Ferrer, Mor Peleg, Enea Parimbelli, Erez Shalom, Carlos Marcos, Guy Klebanov, Iñaki Martínez-Sarriegui, Nick Lik San Fung, Tom Broens

**Abstract**—MobiGuide is a distributed decision-support system (DSS) that provides decision support for patients and physicians. Patients receive support using a light-weight Smartphone DSS linked to data arriving from wearable monitoring devices and physicians receive support via a web interface connected to a backend DSS that has access to an integrated personal health record (PHR) that stores hospital EMR data, monitoring data, and recommendations provided for the patient by the DSSs. The patient data model used by the PHR and by all the system components that interact in a service-oriented architecture is based on HL7's virtual medical record (vMR) model. We describe how we used and extended the vMR model to support communication between the system components for the complex workflow needed to support guidance of patients any time everywhere.

## I. INTRODUCTION

Providing clinical decision-support (CDS) to patients in addition to care providers is a difficult challenge. In the MobiGuide project ([www.mobiguide-project.eu](http://www.mobiguide-project.eu)) we are developing a Ubiquitous Guidance System that provides clinical-guideline-based decision support to physicians through web interfaces and to patients via Smartphone interfaces. Guidance can be made available *anytime and everywhere* by employing a distributed decision-support system (DSS), which includes a backend DSS that has access to the full guideline knowledge base and the complete personal health record (PHR) and a light-weight mobile DSS (mDSS) that runs on the patient's Smartphone and has access to signal data collected from a body-area network (BAN) of wearable sensors. The PHR includes in addition to the BAN data, clinical data coming from hospital electronic medical records, recommendations delivered to the patients by the distributed DSS, abstractions or patterns found in the data by the DSS and responses provided to DSS recommendations by users (patients and physicians).

In order to improve chronic patients' management by the MobiGuide system, we need to assess the patient's current condition from BAN signal data (e.g., physical activity level

This paper has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no. 287811.

A. González-Ferrer, M. Peleg and G. Klebanov are with University of Haifa, Israel (+972-4-828-8509; fax: +972-4-828-8522; e-mail: morpeleg@is.haifa.ac.il), E. Parimbelli is with U. of Pavia, Italy (e-mail: enea.parimbelli01@ateneopv.it), E. Shalom is with Ben-Gurion University, Israel (e-mail: erezsh@bgu.ac.il), C. Marcos is with ATOS, Spain (e-mail: carlos.marcos@atosresearch.eu), I. Martínez-Sarriegui is with U. Politécnica de Madrid, Spain (e-mail: imartinez@gbt.tfo.upm.es), N.L.S. Fung is with U. Twente, The Netherlands (e-mail: l.s.n.fung@utwente.nl) and T. Broens is with MobiHealth B.V., The Netherlands (tom.broens@mobihealth.com).

determined from pedometer and accelerometer data, blood glucose measurements determined by glucometers with Bluetooth connections), supplemented by data that is proactively reported by patients (e.g., patient with gestational diabetes mellitus (GDM) reporting eating extra carbohydrates at a wedding). This changes the "traditional" workflow where interaction with the patients is usually limited to periodic face to face encounters with the caregiver. Instead, in MobiGuide, data collection by the BAN, proactive data reporting by patients, and delivery of clinical recommendations to patients and physicians is continuous.

A major challenge in developing clinical DSSs is their interoperability with electronic health records (EHR), which can be facilitated by standards [1]. In MobiGuide the complexity of interoperability increases, as the system includes over twenty interacting components and a wide range of data sources and types. Targeting the interoperability challenge, MobiGuide's integrated PHR uses the HL7 virtual medical record (vMR) standard [2] as a conceptual model for storing patient data. In this paper we show how this standard is also used for the communication flow and coordination between system components, extending it appropriately to address the management of the distributed clinical workflow, which includes also the patient.

## II. METHODS

### A. Standards for CDSS: HL7 Virtual Medical Record

We carried out a review of possible standards to be used for the patient data model. We decided that the conceptual model of the HL7 vMR could be very appropriate [2], mainly due to its relatively small set of classes which simplifies its learning curve and the time taken to represent different data items. The model is built upon two axes, one to represent the type of clinical information (e.g. Procedure, Observation, Problem, Substance Administration, Goal, Encounter) and a second one related to the workflow moment (e.g. Proposal, Order, Event). Its good documentation and the fact that it was the only standard to our knowledge designed specifically for decision support made it the best candidate to use in MobiGuide. In a second stage, we carried out an analysis of how to use the standard to represent a complete existing database that included hospital EHR data for one of the domains used in the project (Atrial Fibrillation), to apply in practice the theoretical review of the first step and to detect cases where the standard lacked the complete semantics needed in order to represent the full data set.

### B. Workflow analysis

In order to understand how the vMR model could be used as a mechanism for asynchronous communication among

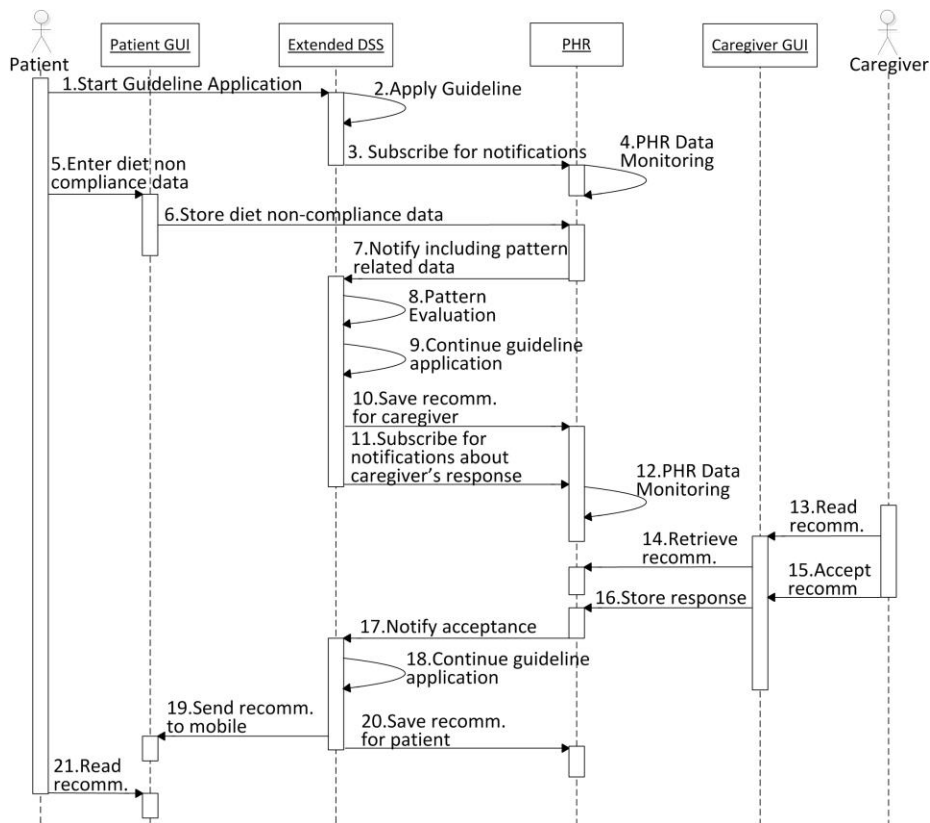


Figure 1. Time sequence diagram for “triggering recommendations for 7 days non-compliance”

system components, we studied in a third stage sample interactions from a set of clinical scenarios from a GDM guideline for which we generated sequence diagrams. In these sequence diagrams we considered which parameters should be passed in messages between interacting components and whether some of these messages could instead be stored as vMR instances in the patient's PHR. In this way, the PHR would store a record of important events – enabling future record data mining- and would also enable asynchronous communication.

In Figure 1 we provide a simplified view which abstracts away from the many different functional components, and lumps together in the "extended DSS": the backend decision-support engine, the knowledge base, the temporal abstraction component, the component that maps guideline knowledge to patient data, and the data notification component. In addition, we lump together within the “PatientGUI” the basic graphic user interface (mobile GUI) and the mDSS.

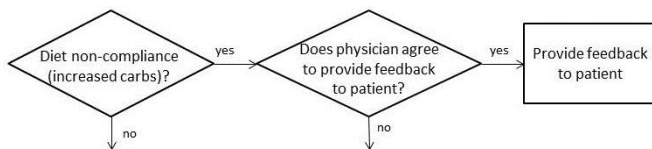


Figure 2. Part of the GDM guideline

The scenario for Figure 1 is taken from the domain of GDM. It concerns guideline recommendations to patient's non-compliance with recommended diet, shown schematically in Figure 2. Figure 1 shows the entire process from the time a patient uses her mobile to indicate an event of non-compliance with diet, and how the system manages this as part of a collection of previous diet non-compliance events

in order to detect a clinically relevant pattern ( $\geq 2$  non-compliance events in 7 days), triggering subsequent evidence-based recommendations to physicians and patients.

### III. RESULTS

Table I contains our analysis of which vMR classes, and their attribute values should be used in order to support the scenario shown in Figure 1. We explain Figure 1 and Table I together, referring between brackets to steps numbered in Figure 1. In the scenario, the extended DSS monitors for "patient non-compliance pattern" by saving into the PHR a specification of the non-compliance event that needs to be detected for the pattern to be calculated [step 3]. This event is an instance of the *ObservationResult* vMR class whose focus is “increased carbohydrates” [Seq #3 in Table I]. Monitoring for this event is done by a notification system that continuously queries for the expected data item (in this case the *ObservationResult*) in the PHR [step 4]. The extended DSS automatically generates the query and when any component stores a matching data item in the PHR a notification event is sent back to the extended DSS. This is shown in [step 6] and Seq #6 of Table I, where the PatientGUI stores the event of increased carbohydrates reported by the patient. In our scenario, the stored event of increased carbohydrates was a second event during the past week, hence a non-compliance pattern has been detected by the extended DSS. In Figure 2, which shows the clinical algorithm, we have answered "yes" to the first decision step.

Clinical recommendations generated by MobiGuide's extended DSS can be either directed at caregivers or at patients themselves. In [step 10] the extended DSS generates a recommendation to the caregiver asking if he wants to

send a feedback message to his patient, to remind her about the importance of compliance to the prescribed diet. As shown in the second decision step in Figure 2 and in Seq #10 of Table I, an instance of the *ProcedureProposal* vMR class representing this recommendation is stored on the PHR by the extended DSS. In addition, the positive response from the physician is awaited (monitored).

TABLE I. ADAPTIONATIONS OF vMR

Seq #	Use Case	Use of vMR in the Scenario
3	The extended DSS monitors for patient non-compliance to diet	ObservationResult focus: Increased carbohydrates (123995008) value: +/++
6	Non-compliance to diet stored in PHR	<b>GL_ID (extension):</b> 201
<b>Propose-order-event/result Workflow Pattern</b>		
10	The extended DSS stores recommendation to the physician to provide feedback message to his patient	ProcedureProposal procedureCode: Notification (C0422202) target: Physician (C0031831) originalText: "The patient didn't follow compliancy recently. Consider sending him/her the following recommendation message: Remember that it is very important that you comply to diet recommendations and blood glucose measurements schedule" <b>DSS_ID (extension):</b> 111 <b>GL_ID (extension):</b> 201
11	The extended DSS monitors for physician's acceptance	ProcedureOrder procedureCode: Notification (C0422202) target: Patient (C0030705) originalText: "Remember that it is very important that you comply to diet recommendations and blood glucose measurements schedule"
16	Physician stores in the PHR his agreement to provide feedback to the patient	<b>DSS_ID (extension):</b> 111 <b>GL_ID (extension):</b> 201
20	Extended DSS stores the message that has just been delivered to the patient on his Smartphone	ProcedureEvent procedureCode: Notification (C0422202) target: Patient (C0030705) originalText: "Remember that it is very important that you comply to diet recommendations and blood glucose measurements schedule" <b>DSS_ID (extension):</b> 112 <b>GL_ID (extension):</b> 201

After storing the *ProcedureProposal* in [step 10], this new recommendation is then retrieved from the PHR by the caregiverGUI [steps 13-14] and the acceptance of the recommendation [steps 15-16] by the physician is stored as a workflow event in the PHR using *ProcedureOrder* vMR class [Seq #16 of Table I].

**Maintaining a link between the recommendation and its acceptance** allows the data notification system described above to be triggered as soon as the caregiver's acceptance (represented in this case by a *ProcedureOrder* [step 17]) enters the PHR. This avoids confusion with other similar previous recommendation records that could exist in the PHR, improving performance too. This linkage is not part of the vMR model but fortunately, the HL7 vMR model includes a standard way to extend itself with attributes of any possible HL7 data type. Hence, the linkage is stored by tagging the recommendation with proprietary IDs that the

extended DSS uses internally to identify recommendations and the guidelines from which they originated (DSS\_IDs and GL\_ID in Table I). The DSS\_ID will be used by the interacting components when saving their reaction in the PHR. For the use case shown here, the caregiverGUI will extract the DSS\_ID from the *procedureProposal* instance (previously generated by the extended DSS [Seq #10 in Table I]) and it will include this DSS\_ID within the subsequent *procedureOrder* instance that it has to save upon acceptance of the physician [Seq #11, 16 in Table I].

Once the extended DSS is notified that the caregiver has accepted the recommendation [step 17] it resumes the guideline execution, generates the recommendation directed at the patient and stores it in the PHR [step 20] as *ProcedureEvent* [Seq #20 of Table I]. The PatientGUI notifies to the patient that a new recommendation has arrived and she can read it [step 21] at any moment.

The knowledge of which vMR classes and vocabulary codes are associated with each guideline recommendation and patient data resides in a knowledge base to which is part of the extended DSS. The mDSS (part of the PatientGUI) receives relevant parts of this knowledge through projections from the backend DSS.

TABLE II. USE OF vMR IN THE GDM SCENARIO

VMR Class	Scenario	
	Times Used	%
ObservationProposal	6	11
ObservationResult	6	11
ProcedureProposal	7	13
ProcedureOrder	6	11
ProcedureEvent	23	42
UndeliveredProcedure	7	13

This scenario was created using a generic design and adapts itself to any kind and type of recommendation defined inside the MobiGuide system. The mechanism shown in Table I [steps 10-20] includes the **workflow pattern of Propose-Order-Event** related to a procedure (sending a recommendation to a patient). The same mechanism can also be applied to substance administrations, encounters, or observations. Simpler patterns can also be used, such as **Proposal-Result**, for example, when the DSS recommends that the patient should measure her fasting's blood glucose level at a certain time. In this case, The DSS issues an *observationProposal* and waits for *observationResult* without requiring an "order" step.

Table II reports the number of times that different vMR classes were used in recommendations of the GDM guideline and their percentage. Moreover, the technical solution that we show relates to a concrete DSS engine and it is demonstrated in a particular guideline, but could be adapted to other architectures and clinical domains easily.

#### IV. DISCUSSION

The important need to give timely decision support as part of the clinical workflow has been already recognized in the

literature [3], [4]. The workflow of asynchronous CDSSs such as MobiGuide can be very complex due to the requirement of supporting decisions outside clinically-controlled environments any time. MobiGuide reminds patients of activities that they should be doing (e.g., measuring blood glucose and ketonuria, exercising, injecting insulin) at a schedule personalized to the patient's routine. At the same time, MobiGuide reacts to asynchronous events such as patterns discovered dynamically in patient biosignal data collected from monitoring devices and to patient input at unforeseen times. This complex workflow raises new interoperability issues for distributed CDSS while keeping compatibility with the common need to link a back-end DSS with the patient record using standards. In this paper we described how we used the vMR standard, which was developed for CDS, to store patient data and for asynchronous communication between system components based on a notification mechanism.

### Adaptation of the vMR schema

We tried to use the vMR “as is”, in order to capture all the different types of data that we want to record in the PHR, including DSS recommendations regarding the patient and responses and acknowledgement from the patient and her physician. Most of the semantics could be captured by the original vMR model. The required adaptation for the MobiGuide system are detailed below and summarized in Table III, along with the number of times each adaptation was used in the GDM guideline, for those adaptations for which we have complete data at present.

In some cases, we were able to capture detailed semantics by relaxing the definitions of vMR classes and their attributes while remaining in the spirit of their intended meaning. For example, using an instance of *ProcedureEvent* was initially thought to be used for clinical procedures (e.g. “knee surgery”) happening, while we adopted it to also represent new scenarios like the event of sending a recommendation message to the patient mobile, which while clinically meaningful in our ubiquitous environment might not be available in medical vocabularies. Similarly, we used an instance of *observationOrder* with ‘dataSourceType=DSS’ to indicate that the recommendation provided does not need physician’s confirmation.

In cases of data items that focused on very specific concepts needing more expressivity than provided by the attributes of the vMR class used (e.g., blood glucose measured after lunch and entered automatically by a specific glucometer that the patient used), we utilized existing vMR attributes for capturing parts of the semantics. For example, the method of data input (manual vs. automatic) was captured by the “method” attribute and the glucometer type was captured by the *datasource\_type* attribute. The “focus” attribute captured the detailed type of blood glucose (BG after lunch) using the post-coordination mechanism of SNOMED [5] (see example in Table III, row 3).

In other cases where we also lacked expressivity in SNOMED itself or in its post-coordination mechanism we

relied on the vMR's extension mechanism to store new attribute-value pairs. For example, we saved an identifier to represent an atrial fibrillation (AF) event detected by the *Linker AF* detection algorithm.

TABLE III. ADAPTATION OF THE vMR SCHEMA

Adaptation	#	Example
Relaxing the definition of vMR classes and attributes	21	Using <i>ProcedureEvent</i> for recording DSS sending a recommendation to the patient through her mobile phone
		<i>observationOrder</i> with <i>dataSourceType</i> =DSS indicates that the recommendation does not need physician’s confirmation
Using several attributes to capture the observation's topic	13	focus: blood glucose method (of data input): manual   automatic <i>datasource_type</i> : glucometer type
Post-coordination of concepts to capture detailed concept focus	4	Blood glucose (BG) level after lunch: focus: <i>271063001 Lunch-time-BG level  :362981000 Qualifier value = 24863003 Postprandial </i>
Gesher_ID: a proprietary ID that stores a code from the Gesher knowledge base for a concept that cannot be composed based on controlled vocabulary terms	*	Gesher_ID = 10009 records an Atrial fibrillation (AF) event detected by the <i>Linker AF</i> detection algorithm focus: code for AF event The focus stores approximate meaning
relatedEntity used to express the target of a recommendation	56	Reminder to measure BG issued by the DSS and delivered directly to the patient without requiring physician approval: <i>relatedEntity</i> (Patient targetRole: code for concept "subject of information" (see Figure 3)
Added attributes to capture workflow: guideline and guideline step that issued a recommendation	56	Added GL_ID and DSS_ID
Added attributes to enable the DSS to address quality of data	4	Added different quality of data attributes, such as "accuracy"

\* work in progress; exact number will be known till the end of May

In other cases, we used other extension mechanisms provided by the *relatedClinicalStatements* and *relatedEntities* classes of the standard. For example, to record the fact that a reminder to measure blood glucose was issued by the DSS and delivered directly to the patient without requiring physician approval for each reminder, we used an instance of *ObservationProposal* whose *dataSourceType* value was “DSS”. To express the target of the recommendation (the patient) we included a *relatedEntity* representing the patient as shown in Figure 3.

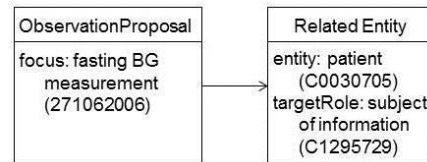


Figure 3. Use of Related Entity to specify the patient as information target of an observationProposal to measure Fasting Blood Glucose

In some cases (last 2 rows in Table III), we had to extend the vMR (via its extension mechanism) to include attributes

whose semantics could not be captured by existing vMR attributes. For example, due to the complex workflow that we were trying to record in the PHR, and facilitated by the semantics of the Asbru [6] language and its DSS execution engine, it was important for us to record in the PHR which guideline step from which guideline issued a concrete recommendation (recorded as a vMR class instance). This is especially important when different guideline steps could issue similar recommendations but we want the DSS to be notified of a recommendation's response related to a particular guideline step, in order to follow the workflow. Hence, as shown in Table I, we added the DSS\_ID and GL\_ID attributes to vMR classes.

Another extension that we added (not shown in the scenario) stems from MobiGuide's requirement to include Quality of Data (QoD) information relating to data acquired from BAN monitoring devices. Such quality indicators, such as the accuracy level, are used by the mDSS to decide whether to repeat measurements, ask for additional input, and which decision options should be recommended.

### Related Work

The vMR is not the only clinical data standard used for CDSS. Another common model is openEHR archetypes [7]. It differs from vMR mainly in its flexibility for creating ad-hoc models; unlike the vMR, openEHR does not provide a small set of classes of data that have predefined structure, but instead allows defining archetypes for each clinical data item, where detailed semantics could be captured that are particular to the data in mind. For example, an archetype to represent 'pain' could capture properties such as the onset of the pain, whether it is intermittent, whether it is spreading, etc. Communities of developers define such detailed archetypes and validate and share them with the community. There is a trade-off between ease of learning of the data model and its flexibility, and we chose the vMR exactly because of its small number of classes with structured predefined attributes. Our reasoning was that when patient data from the EMR and abstractions found in signal data had to be mapped to a standard model, it would not take a long time to represent them in the vMR model, once we develop clear guidelines that assist in such mapping. We are now in the process of developing such guidelines and do not yet have results on the time it takes in practice to represent data items in the vMR model. Yet, an important evaluation that we have accomplished in this paper is to determine that the vMR in practice can be adequately used to represent data items such that CDSS and communication between system components in the notification system could be supported with very little need for extension of the vMR.

A related work is that by Marcos et al. [1] who presented an approach to integrate the PROforma computer-interpretable modeling language [8] with EHRs, where openEHR archetypes, OWL expressions, and SNOMED terms are used to describe the relevant medical concepts. That work does not address how they could reuse the conceptual model provided by archetypes for extending the interoperability to a distributed system and they do not use

the vMR standard structure to create the archetypes (as we proposed in [2]), but instead they use directly the archetypes developed ad-hoc by the openEHR community. By using the LinkEHR platform they are able to retrieve EMR data as archetype instances conformant to the selected -and specialized- archetypes, so that using a mediator module able to connect to any XML data source, they feed the PROforma DSS. They too have relied on SNOMED's post-coordination mechanism, but acknowledged the option of specialization of existing archetypes to express more detailed semantics. They specialized archetypes by adding specific attributes for facilitating integration with PROforma decisions (e.g., in order to support a PROforma argumentation rule that excludes patients with severe comorbidities, the original archetype that records a problem encountered during patient evaluation was extended with the attribute 'present' for accounting the presence/absence of the problem and with a numeric field for capturing the comorbidity score).

### Limitations and future work

The work presented here has some limitations; the full guideline is being modeled at the time of writing, so complete validation has not been achieved. However we already analyzed and tested some complex scenarios involving different types of recommendations (procedures, notifications and recommendation that need patient data entries) and our proposal seems to work well. Its generalization to the complete guideline may still raise exceptions that need extra adaptations, but the core of our solution will remain intact.

### REFERENCES

- [1] M. Marcos, J. A. Maldonado, B. Martínez-Salvador, D. Bosca, and M. Robles, "Interoperability of clinical decision-support systems and electronic health records using archetypes: a case study in clinical trial eligibility," *J. Biomed. Inform.*, vol. 46, no. 4, pp. 676–689, 2013.
- [2] A. González-Ferrer, M. Peleg, B. Verhees, J.-M. Verlinden, and C. Marcos, "Data Integration for Clinical Decision Support Based on openEHR Archetypes and HL7 Virtual Medical Record," in *ProHealth12/KR4HC12 (LNAI 7738)*, 2012, pp. 71–84.
- [3] K. Kawamoto, C. A. Houlihan, E. A. Balas, and D. F. Lobach, "Improving clinical practice using clinical decision support systems: a systematic review of trials to identify features critical to success," *BMJ*, vol. 330, no. 7494, p. 765, 2005.
- [4] G. J. Downing, S. N. Boyle, K. M. Brinner, and J. a Osheroff, "Information management to enable personalized medicine: stakeholder roles in building clinical decision support.," *BMC Med. Inform. Decis. Mak.*, vol. 9, p. 44, Jan. 2009.
- [5] J. Pathak, J. Wang, S. Kashyap, M. Basford, R. Li, D. R. Masys, and C. G. Chute, "Mapping clinical phenotype data elements to standardized metadata repositories and controlled terminologies: the eMERGE Network experience," *J. Am. Med. Informatics Assoc.*, vol. 18, no. 4, pp. 376–386, 2011.
- [6] Y. Shahr, S. Miksch, and P. Johnson, "The Asgaard project: a task-specific framework for the application and critiquing of time-oriented clinical guidelines," *Artif. Intell. Med.*, vol. 14, no. 1–2, pp. 29–51, Sep. 1998.
- [7] D. Kalra, T. Beale, and S. Heard, "The openEHR foundation," *Stud. Health Technol. Inform.*, vol. 115, pp. 153–173, 2005.
- [8] J. Fox, N. Johns, C. Lyons, A. Rahmzadeh, R. Thomson, and P. Wilson, "Proforma: a general technology for clinical decision support systems," *Comput. Methods Programs Biomed.*, vol. 54, no. 1–2, pp. 59–67, 1997.