Architecture for a Ubiquitous Context-aware Clinical Guidance System for Patients and Care Providers

Mor Peleg[1], Tom Broens[2], Arturo González-Ferrer[1], Erez Shalom[3]

[1]Department of Information Systems, University of Haifa, Haifa, Israel, 31905; [2]MobiHealth B.V., Enschede, The Netherlands; [3]Medical Informatics Research Center, Department of Information Systems Engineering, Ben Gurion University of the Negev, Beer-Sheva, Israel

Abstract. Traditional clinical guideline-based decision support systems (DSS) provide patient-specific recommendations to care providers during clinical encounters. In the MobiGuide project we extend this functionality to develop a secure guideline-based ubiquitous guidance system (UGS) which provides personalized guidance to patients and shared decision-making, any time and everywhere, while addressing also non-clinical patient context. Guided by these objectives, and following a top-down approach, we developed a generic, distributed, service-oriented architecture (SOA) for UGS. We propose this architecture as a starting point for UGSs in other clinical domains and healthcare settings. We compare our UGS architecture with existing architectures for DSSs.

Keywords: clinical guidelines, ubiquitous computing, software architecture, decision support, patient guidance

1 Introduction

Traditionally, clinical guideline-based DSS provide patient-specific recommendations to care providers during clinical encounters. Such DSSs match clinical guideline knowledge, represented as computer-interpretable guidelines (CIGs), with patient data from an electronic medical record (that we refer to as Personal Health Record, PHR) to provide specific recommendations to care providers. In the MobiGuide project (www.mobiguide-project.eu) we go beyond traditional DSSs by including patients as users and providing them with evidence-based clinical recommendations even when they are outside clinically-controlled environments (any time, everywhere) to provide multiple benefits. First, by involving patients in their healthcare and by personalizing decision-support to their non-clinical context (e.g., patient adherence level, no network connectivity), recommendations become more suitable to the patients, facilitating better treatment adherence. Second, by supplying patients with a mobile Body Area Network (BAN) consisting of wearable bio-signal monitoring devices connected to a mobile device (e.g. smartphone), the patient's condition can continuously be monitored. Coupled with a DSS that can operate on a backend server or on the mobile device, chronic patients can rest assured that based on their real-time condition at their everyday environment, alerts, reminders and clinical guideline-based recom-

adfa, p. 1, 2013.

mendations are provided to them when situations requiring attention arise. And third, the patients' care providers could use the Ubiquitous Guidance System (UGS) system as well, achieving better guideline adherence, by initiating interactive sessions that would provide patient-specific recommendations or by being notified in a data-driven manner when they need to attend to patients whose condition requires their attention.

Figure 1 provides an overview of our vision of UGS. Traditional CIG-based DSSs include i) CIG representation of clinical practice guidelines (CPGs) stored in a Knowledge Base (KB) serving as knowledge sources, ii) PHR storing the patient's data and iii) a DSS that matches knowledge with data to provide patient-specific recommendations to care providers. UGS include the following extensions (a) patients as users; (b) BAN, including wearable biosensors, signal analysis algorithms, a lightweight mobile DSS and a mobile device with the patient UI; different sources of data integrated into the PHR, including: (c) data from multiple hospital Electronic Medical Records (EMRs), (d) signal data and patient input collected via the BAN, and (e) DSS recommendations, temporal abstractions detected from PHR data and data entered by physicians; and (f) personalized context-aware CIG knowledge base (KB).

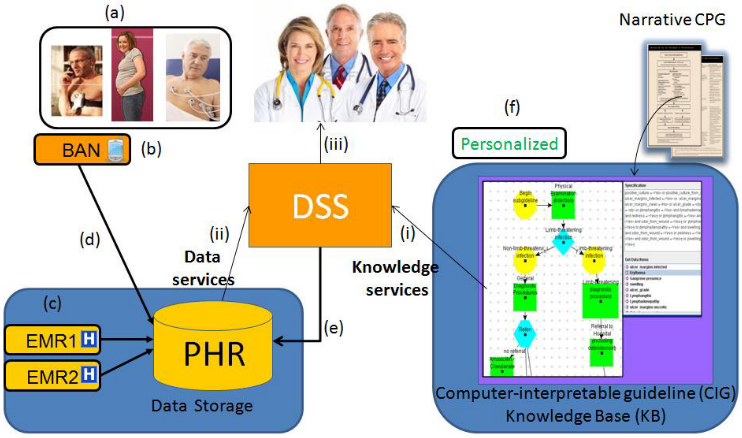

Figure 1. High level overview of a Ubiquitous Guidance System

Transitioning from traditional DSSs to context-aware UGSs for patient and care provider requires careful analysis, translating high-level system goals into functional requirements and designing system architecture capable of meeting the requirements. We describe our analysis process that led to the definition of a functional architecture for UGSs, which we think is generic enough to be adapted and extended for additional domains and healthcare settings. After explaining the architecture and how it meets system goals, we compare it with existing architectures CIG-based DSSs.

2 Methods

The envisioned UGS is highly complex: i) it facilitates a range of different end-users and integrates with third-party components, ii) exhibits complex overall functionality, also at the level of interrelated individual components, and iii) parts of the system will be distributed on different hosts. To deal with this complexity, foresee and react on evolving requirements, reach completeness, and perform risk management, the MobiGuide project adopted an iterative development process [1].

The first iteration aimed to develop a reference architecture to guide future iterations. ISO/IEC/IEEE 42010 defines architecture as: "the fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution." From this definition we derive some focus points: i) identification of components that interact and together constitute the system, ii) specification of the system's environment by identifying its users, their requirements and their interaction with the system and iii) identification of additional design and evolution requirements such as distribution, security, accounting.

We used a top-down, step-wise refinement approach to define the MobiGuide architecture. We identified: i) **external perspective** (known as black-box architecture) that considers the system as a "unified whole", and views it from the perspective of the system's users who want to use it for some purpose. Users can be a human or another computer system. This perspective shows "what" behavior the system is capable of offering; ii) **internal perspective** (known as white-box architecture) that considers the system as a related group of components. It reveals the internals of the system, showing "how" the system is capable of offering the system behavior.

Based on stakeholder analysis, black-box architectures were defined revealing end-users and third party systems. These architectures were further refined to white-box architectures showing generic UGS functions and their interrelations. On lower levels of abstraction it showed MobiGuide specific functional components, component interaction and information flows.

The architecture is used to develop specific designs in each iteration. These designs are developed in SOA to facilitate i) development of independent yet interoperable component by different teams and (ii) integration of third-party components. For the first iteration, we analyzed the involvement of 3 stakeholders (patient, care provider, knowledge engineer) in scenarios taken from a day in the life of an atrial fibrillation (AF) patient who would use MobiGuide. The prototype consists of 20 components developed by 8 distributed development teams. Both architecture and evaluation of the first prototype resulted in a development roadmap to guide future iterations.

3 MobiGuide Reference Architecture

Developing a UGS that provides personalized recommendations to thousands of patients and care providers requires careful consideration on overall system scalability and robustness. Consequently, we adopted the following principles in our architecture: i) presence of a data and control plains, enabling a data driven system using subscribe/notify mechanisms, ii) the server-side backend DSS is the "controller", enabling decision support for different actors on different component of the system at different locations. For example, enabling shared decisions on physicians' desktop applications or interactions with patients and delivery of DSS recommendations on the mobile DSS located on the patient's mobile device, and iii) strict functional separation of concerns between components leading to independent and specialized controllers within components.

Figure Figure **2** shows the overall white-box architecture of the system surrounded by the stakeholder who use system services. **Patients** and **care providers** receive recommendations of the system and can control its behavior. The **knowledge engineering team** is responsible for interpreting the relevant CPG, defining and specifying the CIG knowledge required for the system to operate. The system has the option to enable **researchers** to analyze system-generated data to discover new knowledge. The **system support team** (e.g. helpdesk, administrators) is responsible for keeping the system operational and assisting users. External systems such as third-party **EMRs** also interact with the UGS by exchanging data.

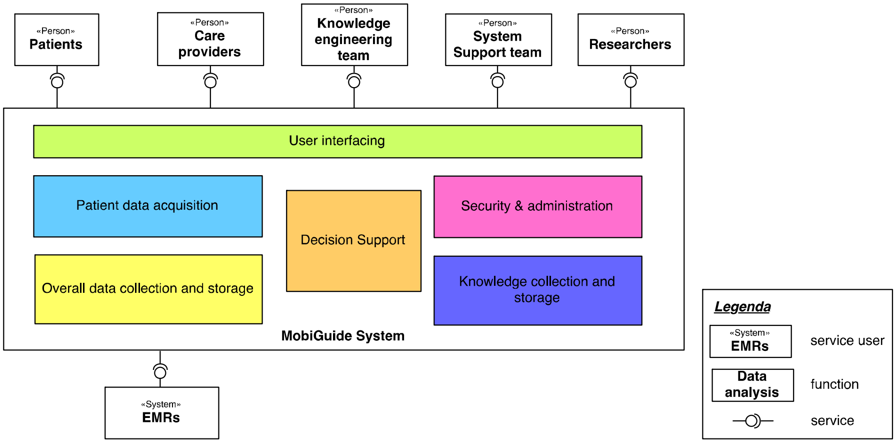

Figure 2. A functional architecture for personalized guidance system, everywhere, any-time

Fig. 2 shows six types of functions essential for a UGS. The **decision support** functions are the heart of the system and include backend and mobile parts. The backend DSS contains functions for sending recommendations to end-users and controlling the behavior of components, based on specified CIGs and collected/generated data. For example the backend DSS is responsible for projecting part of the knowledge to the mobile DSS so that it can provide patients with decision-support independently, with supervision of the backend DSS. While the mobile DSS has no access to PHR data, the projected knowledge captures the patient's current state. This is especially useful when the user loses Internet connection to the DSS backend. **User interfacing** functions are required to provide a presentation layer to stakeholders. These can be offered in different modalities, such as desktop, web and mobile device GUIs. **Patient data acquisition** functions are required to gather data from the patients and feed them into the system. The BAN enables mobile collection of vital signs and/or manual data input by the patient. **Overall data collection and storage** functions are required to uniformly store and expose data to consuming components and integrate third-party EMRs. Semantic data integration is performed using common clinical data standards [2]. **Knowledge collection and storage** are required to store formalized knowledge, perform knowledge-data mappings and perform temporal abstraction on data. Finally, **security and administration** functions are needed to comply with privacy and security legislation and perform administration tasks.

The first iteration focused on the AF disease domain and realized a subset of the overall functionality, including: i) GUIs for patients, care providers and knowledge engineers, ii) shared decision making between patients and care providers, iii) integration of backend DSS with PHR data storage, BAN and temporal reasoners to generate abstractions derived from bio-signals that are relevant to physicians, and iv) initial semantic data integration combining data from hospital EMRs, BAN sensors, DSS recommendations, and user input, while maintaining basic security and privacy.

Requirements that will be addressed in later iterations include comprehensive privacy and security, provide a mobile DSS operating on the BAN's Smartphone, context-aware knowledge, and supporting additional users including system support team, and researchers, who will be able to use intelligent data analysis algorithms.

4 Discussion

Our Decision Support Services [3] includes traditional services such as *Knowledge Services* (Figure 1(i)) that connect the DSS to the CIG KB, and *Data Services* (Figure 1(ii)) which are used to retrieve the data from the PHR. We used other traditional services such as *Data Analysis Services* to analyze temporal patterns and *Logging Services* to store the DSS trace, which are part of our backend DSS, PICARD [4]. We also added novel services compared to those identified in [3] such as: 1) *Signal Processing Service*s to analyze the data come from the BAN; 2) *Shared Decision Services* to allow common decision of physician and the patient; 3) *Compliance Checking Services* to fetch patients' response to UGS recommendations; and 4) *Knowledge to Data Mapping Services* which map CIG knowledge to PHR data. In our view, this set of services enables MobiGuide's ubiquitous functionality as a SOA. The scalability and extensibility of the system would need to be evaluated after implementation of all planned components is completed.

Note that the two main criticisms commented in [3] about traditional non-SOA architecture driven by CIG formal knowledge and corresponding execution engines are addressed in our solution. The first issue –"unlikely consensus of using a formal standard CIG language"—is addressed in our architecture by allowing different CIGs to be plugged in, since the DSS engines are required to follow a SOA approach. The second issue –"challenges related to the local mapping of clinical concepts and data"—is enhanced by the SOA knowledge-data mapper [5], one of the knowledge collection and storage subcomponents of our architecture, which bridges semantic gaps between CIG concepts and PHR data. This component can be also linked with the CIG engine selected, improving the extensibility of our approach.

As patients become more involved in their health, clinical guideline-based DSSs will guide not only care providers and will be used outside clinically-controlled environments. This shift will result in emergent UGSs for patients and care providers, such as MobiGuide, where the DSS itself is only part of a wider architecture that includes different interacting components that use diverse health information to achieve further goals. We propose that developers of future UGSs could leverage our novel functional architecture that is generic and extensible and was derived from stakeholder-based and scenario-driven analysis. UGS developers could leverage by following this architecture while substituting some of the components by others as long as they are functionally similar (e.g., sensors, KBs, decision modules, patient-data sources) and extending it to cover new research topics (e.g., to account for patient comorbidities, care teams, longitudinal data mining of PHR data, etc.).

A recent review of CIG-based DSSs [6] focuses on a smaller number of functional requirements of these applications, addressing management of CIG knowledge, its connection to data, and use of standards. The architecture for UGSs we presented is more complex and could be supported using SOA. Within this paradigm of clinical DSS the most important functional component identified in [3] is a so-called DSS controller module—software that orchestrates the calls to clinical information system (CIS) services and external DSS services. Our proposed architecture does not represent the controller as a unique separate component. Instead, we have small controllers embedded in the developed services (managing their own messaging) and a coordinator of the process (a virtual controller) — the back-end DSS. This way, our architecture is able to manage two different plains, control flow and data flow (as explained in Section 3), maintaining an asynchronous and distributed behavior. We could refer to our architecture as *virtually controlled three-tier SOA*. The traditional three-tier application development separates aspects of the application into presentation, business logic, and data access, and our architecture does similarly (Figure 2); but instead of focusing on the application level we focus on an enterprise architecture level, building the various tiers by encapsulating functionalities in different web services. The PHR is a central data storage component, affecting how the services and capabilities of the system are organized, but actually the control resides in services and is coordinated by PICARD DSS [4], that orchestrate the process using the CIG knowledge base.

Acknowledgements. This study was part of the MobiGuide project partially funded by the European Commission 7th Framework Program, grant #287811.

References

1. Hoffer JA, George JF, Valacich JS. Modern Systems Analysis and Design, 2005;
2. González-Ferrer A, Peleg M, Verhees B, Verlinden J-marc, Marcos C. Data integration for Clinical Decision Support based on openEHR Archetypes and HL7 Virtual Medical Record. In: LNAI 7738: Joint ProHealth / KR4HC Workshop. 2012:71-84.
3. Kawamoto K, Jacobs J, Welch BM, Huser V, Paterno MD, Del Fiol G, et al. Clinical Information System Services and Capabilities Desired for Scalable, Standards-Based, Service-oriented Decision Support: Consensus Assessment of the Health Level 7 Clinical Decision Support Work Group. Proc AMIA Symp. 2012:446–55.
4. Shalom E, Fridman I, Shahar Y, Hatsek A, Lunenfeld E. Towards a realistic clinical-guidelines application framework: desiderata, applications, and lessons learned. LNCS 7738: Process Support and Knowledge Representation in Healthcare.2013:56-70.

5. Peleg M., Keren S, Denekamp Y. Mapping Computerized Clinical Guidelines to Electronic Medical Records: Knowledge-Data Ontological Mapper. J Biomed Inform 2008;41:180-201
6. Isern D, Moreno A. Computer-based execution of clinical guidelines: a review. International journal of medical informatics 2008;77(12):787–808.