Guideline-based Decision Support for the Mobile Patient Incorporating Data Streams from a Body Sensor Network

Nick L. S. Fung, Valerie M. Jones, Richard G. A. Bults and Hermie J. Hermens
University of Twente, 7500 AE Enschede, The Netherlands
Email: l.s.n.fung@utwente.nl

***Abstract*—We present a mobile decision support system (mDSS) which helps patients adhere to best clinical practice by providing pervasive and evidence-based health guidance via their smartphones. Similar to some existing clinical DSSs, the mDSS is designed to execute clinical guidelines, but it operates on streaming data from, e.g., body sensor networks instead of persistent data from clinical databases. Therefore, we adapt the typical guideline-based architecture by basing the mDSS design on existing data stream management systems (DSMSs); during operation, the mDSS instantiates from the guideline knowledge a network of concurrent streaming processes, avoiding the resource implications of traditional database approaches for processing patient data which may arrive at high frequencies via multiple channels. However, unlike typical DSMSs, we distinguish four types of streaming processes to reflect the full disease management process: Monitoring, Analysis, Decision and Effectuation. A prototype of the mDSS has been developed and demonstrated on an Android smartphone.**

Keywords*—Telemedicine, decision support systems, software design, body sensor networks, pervasive computing.

I. Introduction

Clinical guidelines are widely regarded as offering much potential to improve the quality of clinical decision making and thereby patient care [1]. To support their implementation in daily clinical practice, many approaches have been developed for formalising and automating clinical guidelines; examples include DeGeL, GLEE and SAGE [1]. In the MobiGuide (MG) project, we aim to extend beyond existing guideline-based decision support systems (DSSs) by developing a pervasive patient guidance system which provides guideline-based recommendations to the patient in a free-living setting as well as to clinicians in a hospital setting [2].

The MG system incorporates, alongside other components, a mobile clinical decision support system (mDSS) that can operate autonomously on the patient's body sensor network (BSN), specifically on his or her smartphone, and provide health guidance to the patient anytime and anywhere. The mDSS receives the guideline knowledge relevant for that patient from a back-end DSS and applies it to the data arriving from the BSN, generating recommendations for the patient when appropriate. Furthermore, the mDSS collaborates with the back-end DSS by sending its results of analysing patient data, which in some cases will be processed further to produce additional guidance for the patient and the clinicians.

Like other conventional knowledge-based systems (KBSs), guideline-based DSSs generally contain a knowledge base for storing the clinical guidelines as well as an inference engine to reason with the stored knowledge; such separation of concerns enables different knowledge to be plugged into the system, thus permitting it to be independent of any particular domain [3]. Furthermore, to provide decision support specific to a particular patient, guideline-based DSSs have typically been designed to query a central database, such as an electronic health record, for the relevant patient data.

However, in pervasive healthcare, data is best modelled as streams instead of persistent relations as the data items may be generated continually at high frequencies and via multiple channels. The BSN of the INTERACTION system for stroke patients, for example, includes 14 3D inertial measurement units outputting data at a total rate of 7.66 kB/s [4], and as noted by Babcock et al., traditional database approaches are not designed to process such data streams [5]. Indeed, this issue becomes more acute when the high data throughput requirements of the Internet of Things are considered.

Therefore, to address the systems challenges of processing potentially large data volumes continually on a smartphone, we base the design of the mDSS on existing data stream management systems (DSMSs), in which streaming processes accept, operate on and output data streams directly. However, we also retain the clear separation of knowledge and reasoning in typical KBSs, thus preserving the advantage of genericity. Further requirements which are also considered include the ability to support the whole disease management process, to pause and resume the reasoning process in case the smartphone is, for example, switched off, and to support scenarios in which patient input is necessary to complete a reasoning process.

Section II presents the design of the mDSS, followed by a discussion in Section III and conclusions in Section IV. Although the mDSS is designed to be domain-independent, we use simple illustrative examples drawn from the two MG pilot domains: atrial fibrillation (AF) [6] and gestational diabetes mellitus (GDM) [7].

II. Design of the mDSS

A. Overall Architecture

Fig. 1 shows the high-level architecture, i.e. set of principal design decisions [8], of the mDSS. Solid rectangles depict system components, each of which represents a subset of the system's functionality, whilst dashed rectangles depict system connectors, i.e. interactions between components. Each

MOBIHEALTH 2014, November 03-05, Athens, Greece

DOI 10.4108/icst.mobihealth.2014.257420

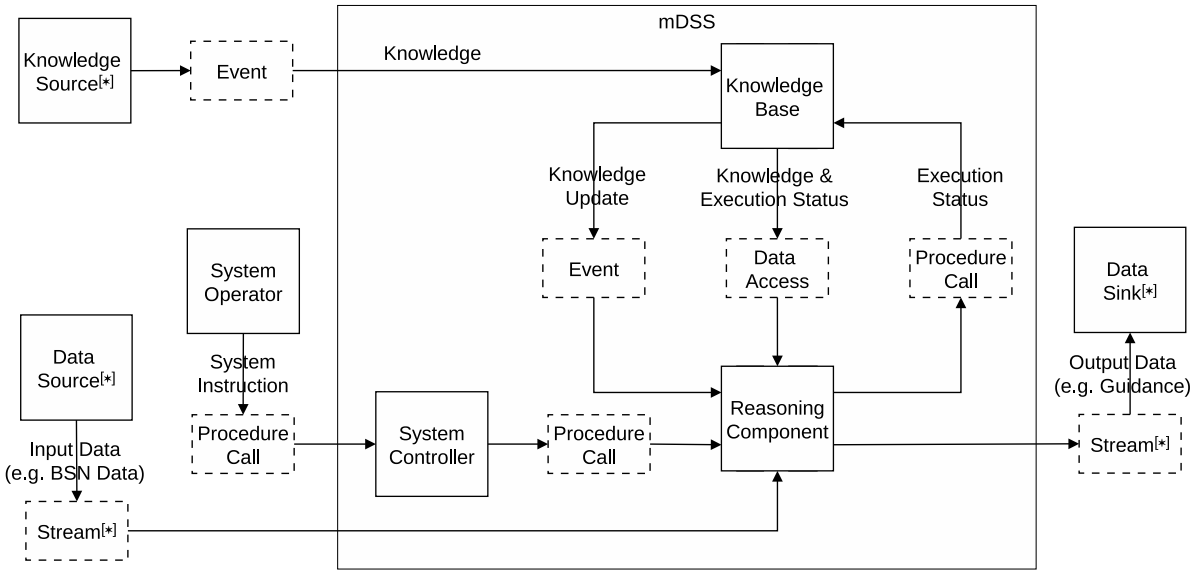

Fig. 1. Overall architecture of the mDSS, abstracting away from the specific details of the MG system. [*] is used to denote multiplicity in the figure.

connector is associated with an arrow representing data flow and is classified using the taxonomy by Taylor et al. [8].

Similar to existing guideline-based DSSs, the mDSS provides decision support by applying the provided guideline knowledge to the available patient data. However, as shown in Fig. 1, our mDSS does not retrieve the input data by querying a database via a data access connector. Instead, the input data are streamed from multiple sources, such as the smartphone GUI and the sensors on the BSN, and are processed directly by the mDSS. Likewise, the mDSS outputs the results of its analysis in multiple streams to its data sinks, which may include the smartphone GUI and any actuators in the patient's BSN.

Furthermore, unlike traditional KBSs in which the inference engine operates directly on the knowledge base [8], the reasoning component of the mDSS is stateful and embodies the guideline knowledge during run-time. Whenever the mDSS is updated with new knowledge, the knowledge base notifies the reasoning component and triggers it to adjust its behaviour by retrieving and assimilating the relevant knowledge as determined by the current state and context of the patient. Likewise, automatic updates may also be performed by the reasoning component as the disease management process progresses.

Thus although the mDSS also contains a knowledge base like other KBSs for storing the received knowledge persistently, its primary function is to allow the pause and resumption of the guideline execution process. The patient may terminate the mDSS by, for example, switching off the smartphone, and in such cases, the reasoning component would be triggered by a procedure call from the system controller to store its execution status. In this way, the reasoning component can resume its operations on restart by simply accessing the knowledge base to retrieve the stored state.

B. Internal Architecture of the Reasoning Component

Similar to the query plans of various existing DSMSs such as Aurora [9] and STREAM [10], our approach for reasoning on data streams is to realise the guideline knowledge as a directed, acyclic network of concurrent, streaming processes. We consider each such process as a component that accepts data items from one or more input streams and processes them sequentially in the order of arrival (which may or may not correspond to the order in which the items were generated). The results of the processing are then sent in a single stream to one or more target components, such as external data sinks or other streaming processes in the reasoning component.

However, unlike the typical streaming processes within a query plan in DSMSs, we distinguish four types of high-level streaming processes, namely Monitoring (M), Analysis (A), Decision (D) and Effectuation (E) as shown in Fig. 2, to reflect the different activities involved in the full disease management process. For example, measuring the heart rate of an AF patient may involve some algorithmic processing specific to the sensor used, whilst deducing that the patient should cease exercising may involve a generic inferencing process. Indeed, guideline representation languages generally include constructs for decisions and actions [11], and in our MADE model, which is based on work previously reported [12], further distinctions are made between diagnostic and therapeutic decisions as well as measurement and therapeutic actions.

We define the four types of processes as follows:

- Monitoring (M) is the process of making observations about the patient and his or her context. Conceptually, this process includes sensing as well, but for the mDSS, it mainly involves transforming the measured data from the data source to lower-level concepts about the patient and his or her context.
- Analysis (A) is the diagnostic process of inferring higher-level, more abstract concepts from the monitored lower-level concepts. In line with Seyfang et al. [13], we use diagnosis to include both a single assessment of a patient (e.g. diagnosis of GDM) as well as on-going assessments of clinical status.

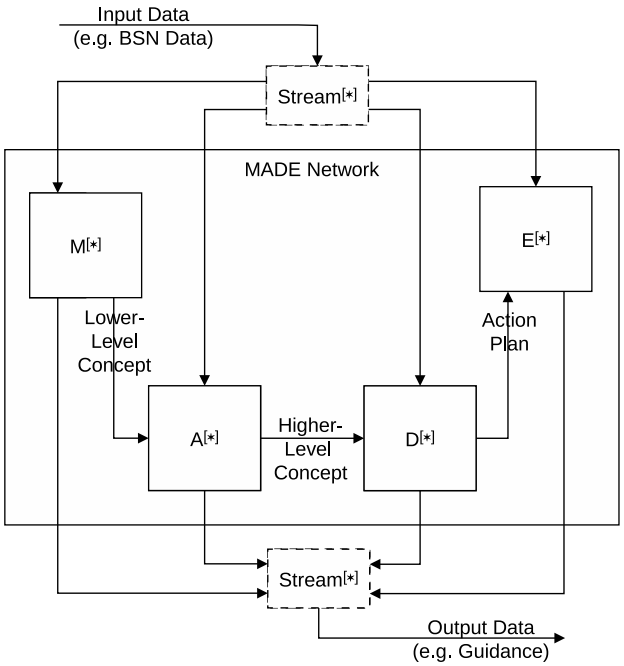

Fig. 2. Architecture of the MADE network in the reasoning component, focusing on the main data flows only. For simplicity, the stream connectors between the MADE processes are also not shown in the figure.

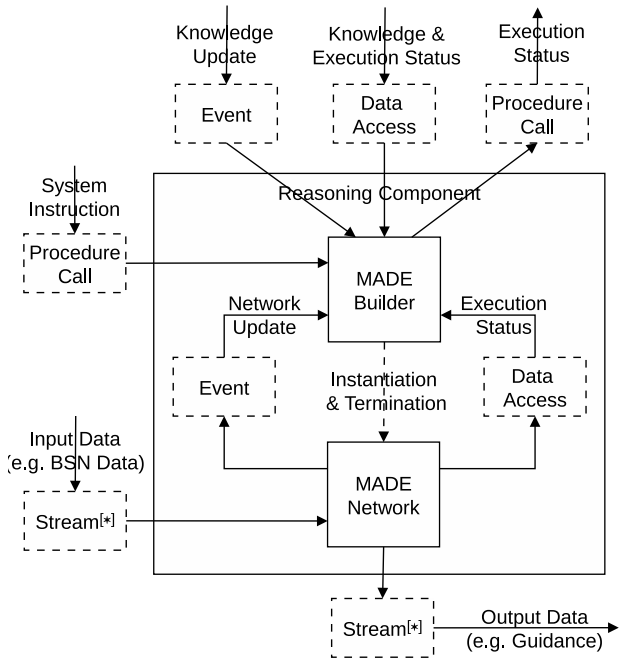

Fig. 3. Internal architecture of the reasoning component. The dashed arrow in the figure denotes the instantiation and termination of the MADE network by the MADE builder.

- Decision (D) is the therapeutic decision-making process of deciding on the appropriate course of action (e.g. to take medication) given the higher-level concepts (clinical abstractions) reached about the patient. Assuming all resources necessary to perform the actions are available, this process is equivalent to a planning process in the AI domain [14].
- Effectuation (E) is the process of performing the decided course of action. Conceptually, this process also involves the patient or actuators performing some physical action, but for the mDSS, it mainly involves instantiating or terminating a MADE process and converting the decided course of action into specific instructions for the (human) actor or the actuators.

As implied by the definitions above and as shown in Fig. 2, M processes can only be succeeded by A processes, and likewise A by D and D by E processes. However, to account for processes which must be performed by the patient or require some manual input to complete, the data flow is not constrained to enter from an M process or exit through an E process. In the MG system for example, as determined by the GDM guideline [7], GDM patients are required to monitor their carbohydrates intake and analyse their own compliance to the recommended diet regime, inputting cases of non-compliance only and thereby stream data directly into the corresponding A process. Furthermore, to improve patient compliance, GDM patients may be reminded regularly to monitor their blood glucose levels, in which case the required output would be streamed directly from an M process to a data sink, namely the smartphone GUI in the MG system.

In addition to the MADE network, the reasoning component contains a MADE builder as shown in Fig. 3 which interfaces with the system controller and the knowledge base of the mDSS. Whenever new guideline knowledge needs to be executed, such as when the patient enters into a new phase of treatment or when the knowledge base is updated by a knowledge source, the MADE builder accesses the knowledge base to retrieve the relevant knowledge and instantiates from it the corresponding network of MADE processes. Furthermore, the MADE builder can access the execution status of the processes in the MADE network, enabling the mDSS to store its execution status when necessary, such as before termination by the system operator.

III. Discussion and Future Work

A. Specification of the MADE Processes

A prototype of the mDSS has been developed for Android smartphones and has been demonstrated as part of the complete MG system for a GDM scenario. In the first prototype, the mDSS receives blood glucose measurements from the smartphone GUI and analyses them to detect any non-compliance to the measurement regime. On detection of non-compliance, the mDSS then notifies the patient via the GUI and sends the results to the back-end for further processing.

Work is on-going to develop and evaluate a fully generic specification of the MADE processes. To ensure its clinical validity in the two MG pilot domains at the minimum, our approach is to induce the specification from the MG requirements and extend it for increased genericity by considering, amongst other factors, the state-of-the-art in AI systems and DSMSs. For example, from the requirement that GDM patients should analyse their own compliance to the recommended diet regime (Section II-B), we can induce that any number of MADE processes may require some form of manual input.

Furthermore, the streaming processes in DSMSs commonly involve sliding windows to evaluate the most recent data only or small summaries to enable incremental data processing [5], [15], and such features may also be incorporated into the MADE specification.

However, as presented in Section II-B, each type of MADE process is different in nature, and of particular consideration is the trade-off between the scope and the complexity of each type of process. For example, the computational complexity of planning and therefore D processes may range from constant time to EXPSPACE-complete depending on the assumptions made, such as whether the permitted actions can exhibit negative effects or not [14].

B. Mechanisms for Quality-Aware Decision Support

Unlike conventional guideline-based DSSs, the mDSS will be used by the patient in a free-living setting, and the inherent noise in such an environment may result in additional problems. For example, an inaccurate blood glucose reading may lead to the erroneous conclusion that a GDM patient is hypoglycaemic and, in turn, cause the patient to consume more sugars than is appropriate for her. In data stream processing, delayed, missing and out-of-order data are also common issues [16], the causes of which may include an unreliable communications infrastructure or malfunctioning sensors.

To account for such problems, we are also investigating an appropriate mechanism for adding quality awareness to the mDSS. Typical examples from the AI domain include the use of probabilistic or fuzzy logics, but for any chosen approach, we believe that it must be adapted for the telemedicine domain. It may not be possible, for example, to accurately quantify the uncertainty in the data, particularly if manual input is required. Furthermore, due to this uncertainty, multiple, mutually exclusive decisions may be recommended at the same time, thus necessitating extra procedures to, for example, select the safest decision to effectuate. Indeed, as part of the MG project, our work on a quality-aware mDSS forms a continuation of the work by Larburu et al. on quality of data, which includes a requirements elicitation methodology for determining its clinical effects [17]. In this way, the clinical validity of the provided decision support may be preserved despite the possible fluctuations in the data's quality.

IV. Conclusion

Traditional guideline-based clinical decision support systems are designed to be used in a clinical environment and to query a central database for the relevant data. In this paper, we have presented the architecture of a mobile DSS that is intended for use by the patient in a free-living setting and is tailored towards the telemedicine domain and the processing of multi-channel data streams such as from a BSN. The design of our mDSS was inspired by existing data stream management systems but extended to account for the different types of activities involved in the full disease management process, namely Monitoring, Analysis, Decision and Effectuation. Furthermore, the clear separation of knowledge and reasoning inherent in typical KBSs is retained, thereby preserving their domain-independence property.

The mDSS currently provides pervasive guidance to the patient by virtue of being implemented on his or her smartphone, but we believe that the MADE approach can be extended further to facilitate pervasive healthcare where processing is distributed across multiple physical devices. Since the MADE processes can, by definition, be executed independently and concurrently on their own threads, they can be distributed arbitrarily across different physical devices provided that the appropriate underlying infrastructure is available. Indeed, similar mechanisms for distributing streaming processes have also been presented to optimise the performance of DSMSs [15].

Acknowledgement

The MobiGuide project (http://www.mobiguide-project.eu/) has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no. 287811.

References

[1] D. Isern and A. Moreno, "Computer-based execution of clinical guidelines: A review," *Int. J. Medical Informatics*, vol. 77, pp. 787–808, 2008.

[2] M. Peleg et al., "Architecture for a Ubiquitous Context-aware Clinical Guidance System for Patients and Care Providers," in *Joint Int. Workshop KR4HC/ProHealth Workshop*, 2013.

[3] W. van Melle, "A domain-independent production-rule system for consultation programs," in *Proc. 6th Int. Joint Conf. Artificial Intelligence (IJCAI'79)*, vol. 2, 1979, pp. 923–925.

[4] B. Klaassen et al., "A System for Monitoring Stroke Patients in a Home Environment," in *Int. Conf. Health Informatics (HEALTHINF 2014)*, 2014, pp. 125–132.

[5] B. Babcock et al., "Models and Issues in Data Stream Systems," in *Proc. 21st ACM SIGMOD-SIGACT-SIGART Symp. Principles of Database Systems*, 2002, pp. 1–16.

[6] A. J. Camm et al., "2012 focused update of the ESC Guidelines for the management of atrial fibrillation: an update of the 2010 ESC Guidelines for the management of atrial fibrillation. Developed with the special contribution of the European Heart Rhythm Association." *European Heart J.*, vol. 33, pp. 2719–2747, 2012.

[7] M. Rigla et al., *Gestational Diabetes Guideline CSPT*, MobiGuide Project (FP7-287811), 2013, Version 1.0, 12/02/2013. Internal document.

[8] R. N. Taylor et al., *Software Architecture: Foundations, Theory, and Practice*. John Wiley & Sons, Inc., 2010.

[9] D. Carney et al., "Monitoring Streams: A New Class of Data Management Applications," in *Proc. 28th Int. Conf. Very Large Data Bases*, 2002, pp. 215–226.

[10] The STREAM Group, "STREAM: The Stanford Stream Data Manager," *IEEE Data Eng. Bulletin*, vol. 26, pp. 19–26, 2003.

[11] M. Peleg et al., "Comparing Computer-interpretable Guideline Models: A Case-study Approach," *J. Amer. Medical Informatics Assoc.*, vol. 10, pp. 52–68, 2003.

[12] T. Broens et al., *Deliverable 2.3: Distribution of Decision Support Functionality*, 2012, available at http://www.mobiguide-project.eu/downloads/category/10-public-deliverables (Accessed July 29, 2014).

[13] A. Seyfang et al., "Combining diagnosis and treatment using ASBRU," *Int. J. Medical Informatics*, vol. 68, pp. 49–57, 2002.

[14] M. Ghallab et al., *Automated Planning: Theory & Practice*. Morgan Kaufmann, 2004.

[15] L. Golab and M. T. Özsu, "Issues in Data Stream Management," *SIGMOD Rec.*, vol. 32, pp. 5–14, 2003.

[16] M. Stonebraker et al., "The 8 Requirements of Real-time Stream Processing," *SIGMOD Rec.*, vol. 34, pp. 42–47, 2005.

[17] N. Larburu et al., "Making Medical Treatments Resilient to Technological Disruptions in Telemedicine Systems," in *IEEE-EMBS Int. Conf. Biomedical and Health Informatics*, 2014.