



KES2014

Application of a conceptual framework for the modelling and execution of clinical guidelines as networks of concurrent processes

Nick Lik San Fung^{a,*}, Ing Widya^a, Tom Broens^b, Nekane Larburu^a, Richard Bults^a, Erez Shalom^c, Val Jones^a, Hermie Hermens^{a,d}

^aUniversity of Twente, 7500 AE Enschede, The Netherlands

^bMobiHealth B.V., 7500 AM Enschede, The Netherlands

^cBen-Gurion University of the Negev, Beer-Sheva 84105, Israel

^dRoessingh Research and Development, 7500 AH Enschede, The Netherlands

Abstract

We present a conceptual framework for modelling clinical guidelines as networks of concurrent processes. This enables the guideline to be partitioned and distributed at run-time across a knowledge-based telemedicine system, which is distributed by definition but whose exact physical configuration can only be determined after design-time by considering, amongst other factors, the individual patient's needs. The framework was applied to model a clinical guideline for gestational diabetes mellitus and to derive a prototype that executes the guideline on a smartphone. The framework is shown to support the full development trajectory of a decision support system, including analysis, design and implementation.

© 2014 The Authors. Published by Elsevier B.V.

Peer-review under responsibility of KES International.

Keywords: Knowledge-based systems; Telemedicine; Decision support; Clinical guidelines; Process modelling; Disease management; Distributed systems; Body Area Networks; eHealth; Evidence-based medicine

1. Introduction

Advances in mobile and wireless technologies and body worn sensors provide an increasingly rich environment for telemedicine services, which are widely regarded as offering huge opportunities for supporting patient empowerment and self-management. Building on past work since 2001¹, we currently apply body area networks for telemonitoring and for providing mobile decision support to patients in the context of a European project MobiGuide, the aim of which is to develop a patient guidance system that gives mobile and ubiquitous decision support to patients with chronic diseases.

In the MobiGuide system, the decision support provided is based on the knowledge in clinical practice guidelines (sometimes referred to as clinical guidelines or just guidelines), which represent the current best clinical practice as

* Corresponding author.

E-mail address: l.s.n.fung@utwente.nl

supported by the latest scientific evidence and which are intended to be used by clinicians as guidance in treating their patients. In the project, we focus on two clinical guidelines for treatment respectively of cardiac patients suffering from a fast and erratic heart beat (atrial fibrillation – AF)² and women who develop diabetes during pregnancy (gestational diabetes mellitus – GDM)³. The MobiGuide system has been prototyped and demonstrated using scenarios based on these guidelines and, when further developed, will be piloted on patients in Spain and Italy.

Since the early days of knowledge-based systems⁴, it has been recognised that separation of concerns – relating to knowledge and the reasoning mechanisms that are applied to knowledge and data – brings the advantage of genericity and enables a plug-and-play approach to knowledge bases. This permits the system to be independent of the application domain, provided that the domain knowledge can be fully represented in the knowledge formalism used. However, for telemedicine applications, we propose that a knowledge-based system should not only be domain-independent, but also device-independent. In other words, a knowledge-based telemedicine system should be independent of the specific physical components that constitute the system, provided that at least one of the physical components (which we will refer to as physical devices to avoid confusion with software components) can function as a knowledge-based system.

Our proposition is based on two observations regarding telemedicine systems. Firstly, the needs of each patient are different, and therefore it cannot be known at design time which exact combination of devices (for example biosensors) is most appropriate for each patient. Secondly, although the back-end infrastructures can be fixed during design-time, the possibility of losing wireless coverage implies that some reasoning must be performed close to the patient on the front-end in order to support ubiquitous decision support. Indeed, the MobiGuide system incorporates, amongst other components, two decision support systems, one on the back-end for complex processing of data and one on a smartphone for providing guidance to the patient anytime, anywhere. Furthermore, with the advent of programmable wearables, such as the Android-based smartwatch, we have the prospect of an even greater number of possibilities for providing distributed knowledge-based decision support to the patient.

As a result, there is a need for the distribution of knowledge and processing across different physical devices at run-time, whether statically during initialisation and/or dynamically throughout the operational phase. Therefore, we have developed a conceptual framework, which we refer to as the MADE framework, for modelling clinical guidelines as networks of concurrent processes. Since each process corresponds to a portion of the guideline knowledge and has, by definition, a direct procedural interpretation, our conceptual framework not only enables the guideline to be partitioned for distribution across multiple devices, but it also allows the reasoning mechanism to be realised as a network of processes. In this way, the distribution of knowledge becomes a means for the distribution of processing.

In Section 2 below we present the state-of-the-art in modelling clinical guidelines and the disease management process in general, and this is followed by a description of our proposed MADE framework in Section 3. Section 4 describes how the framework was applied in the case of a guideline for GDM, and in Section 5 we present a discussion and plans for future work, which is followed by the conclusions in Section 6.

2. State-Of-The-Art

The main formalisms for modelling guideline knowledge have been categorised by Peleg⁵ as:

- Document models, which encapsulate the properties of guideline documents (e.g. their target audiences).
- Decision trees and probabilistic models, which capture the algorithmic knowledge in guidelines.
- Task-network models, which represent clinical guidelines as networks of tasks.

Of particular relevance to our approach are the task-network models as they enable clinical guidelines to be formalised such that the complete disease management process can be automated. Main exemplars of this approach are GLIF, Guide, Asbru, GASTON, GLARE, HELEN, *PROforma*, SAGE and EON. In general, task-network models represent clinical guidelines as hierarchical plans which may contain constructs for decisions and actions as well as embedded sub-plans⁶. The *PROforma* language, for example, distinguishes between Actions, Enquiries (which may be considered as a special type of action⁶), Decisions and Plans (which are collections of other tasks)⁷. Task networks encapsulate the control flow (i.e. the logical ordering) between different processes over time, whether they are to be executed sequentially, iteratively or concurrently⁶.

An alternative approach to modelling clinical guidelines and, more generally, the disease management process is to focus on the data flow between each process instead of the control flow. A typical example is described by Carson et al. in 1998 for the design and evaluation of clinical decision support systems⁸. Their model comprises a sequence of three main processes controlling the patient state: one to monitor the patient, another to make the appropriate clinical decisions and the third to effect the chosen decision⁸. Their model was not formalised to enable automatic execution of clinical guidelines, but it was applied at a conceptual level and was shown to be applicable at many levels of disease management, from the business level to the patient level⁸.

In 2013, Lasierra et al. presented an ontological approach based on a similar model by Kephart and Chess for developing home-based telemonitoring systems that can integrate data from heterogeneous sources⁹. The Kephart and Chess model comprises four processes, namely Monitor, Analyze, Plan and Execute, and was originally proposed in 2003 for use in the computer science domain to represent the self-management of computing systems¹⁰. However, Lasierra et al. demonstrated its applicability in the healthcare domain as well by developing the HOTMES ontology for specifying the requirements of home-based telemonitoring systems in terms of the four different processes⁹.

Space does not permit an extensive state-of-the-art here, for this we refer the reader to the comprehensive review by Peleg⁵ and the resources of OpenClinical¹¹.

3. Description of the MADE Framework

3.1. The MADE model of disease management

Our proposed framework is based on the model of Carson et al. but adapts it for a different purpose, namely for facilitating execution of clinical guidelines. As with their model, and in contrast to most existing approaches to guideline formalisation, our main emphasis is on capturing the data flows between different processes. The data-flow view enables similar tasks for different purposes to be easily identified and grouped together for distribution at run time.

Furthermore, our approach differs from that of Lasierra et al. in two main respects. Firstly, whereas the Monitor and Execute tasks are considered by Lasierra et al. to be the entry and exit points of their system respectively¹², we abstract away from the specific functioning of the computerised system and model the complete disease management process at the knowledge level. Secondly, we do not assume the presence of a central database as implied by Lasierra et al. in their approach to integrate patient data from multiple sources¹². Instead, our framework is based on the streaming model of data processing, in which data arrives continuously in multiple streams and is operated on directly without an intermediate archival step¹³.

In terms of the model itself, we consider the disease management process as consisting of four processes controlling the environment (Fig. 1): Monitoring (M), Analysis (A), Decision (D), and Effectuation (E). Like the model of Carson et al., our model of disease management includes processes for monitoring the patient and performing the associated actions. However, to clearly distinguish between diagnostic and therapeutic decision-making, we have partitioned the clinical decision making process into two separate processes, namely Analysis and Decision. Furthermore, unlike either the Carson et al. or the Kephart and Chess model, our MADE model accounts for the possibility of the system to monitor both the state of the patient as well as the patient's surroundings and to perform actions which affect the patient state indirectly via changes in his or her surroundings or in the function of the system itself.

Thus in our MADE model, which is based on work previously reported¹⁴, we define the four processes (Monitoring, Analysis, Decision and Effectuation) as follows:

- Monitoring (M) is the process of making observations about the environment, including both the internal as well as the external environment of the patient. This involves processing and assigning meaning to physical stimuli, transforming them to low-level concepts (i.e. data) about the state of the environment.
- Analysis (A) is the process of making abstractions from the low-level concepts, transforming them into high-level, more meaningful concepts (i.e. information) about the environment. In the clinical context, Analysis can therefore be seen as a diagnostic process, which may, in the simplest case, only involve a single assessment of the patient but can include on-going assessments as well¹⁵.

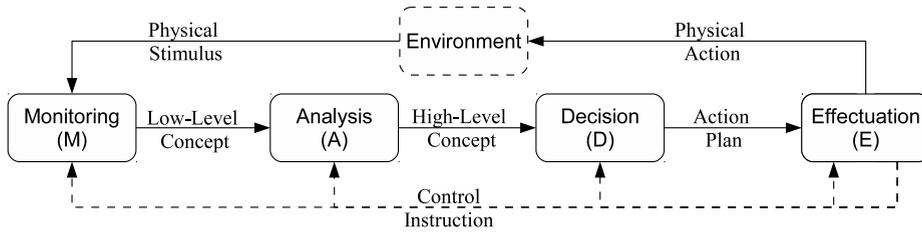


Fig. 1. The MADE model of a generic disease management process. For simplicity, the closure of the MADE loop, i.e. the flow of control instructions and the flow to and from the Environment, will not be shown explicitly in subsequent figures.

- Decision (D) is the process of deciding on the appropriate plan (i.e. course of action) given the current state of the environment and can therefore be seen as therapeutic decision-making. In our model, the resources required to execute the plan, such as sensors and actuators, are assumed to be available. Thus in the strict AI sense, the Decision process also includes some scheduling as well as planning¹⁶.
- Effectuation (E) is the process of performing the decided course of action, with the intention of bringing about a change in the patient’s state, whether directly or via changes in the patient’s external environment. The course of action may also involve controlling another MADE process, such as changing the frequency at which the patient’s state is analysed or the specific physical stimuli that are monitored.

As a simple example of applying the MADE framework, consider the MADE model of a telemonitoring system for the avoidance of symptoms arising from elevated heart rate. Such a system may be of relevance for cardiac patients with atrial fibrillation¹⁴, and conceptually, it requires a Monitoring process to measure the patient’s heart rate, the results of which will be fed into an Analysis process to infer the presence or absence of high heart rate. For this system, the Decision process can be a simple switch; if high heart rate is present, the appropriate course of action, e.g. to sit down and take a rest, will be selected and effectuated.

3.2. Decomposition of MADE processes

A straight-forward approach to achieve distribution is to consider each MADE process in Fig. 1 as a single conglomerate function and distribute whole processes en-bloc, such that the Monitoring process is completely performed on one device, the Analysis process on another or the same device, and likewise for Decision and Effectuation. However, to maximise the potential benefits of distribution, it is useful to consider the ways in which each MADE process can be decomposed successively into smaller sub-processes. In our MADE framework, two types of decomposition are identified: parallel (Fig. 2(a)), where each process is decomposed into a set of concurrent sub-processes; and serial (Fig. 2(b)), where each process is decomposed into a sequence of sub-processes.

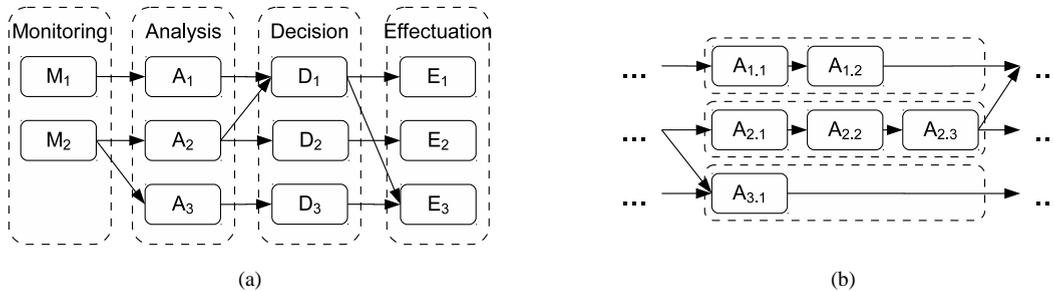


Fig. 2. The MADE model of a generic disease management process after one level of (a) parallel decomposition and (b) serial decomposition. For simplicity, only the Analysis processes are shown in (b).

3.2.1. Parallel decomposition

In the clinical domain, multiple decisions are often made concurrently. For example, the NICE guideline on obesity states that for adult patients, clinical intervention may involve some combination of diet, physical activity, drugs and surgery¹⁷. Although separate from each other, these decisions will likely be made during one consultation and may require a common set of patient information (i.e. high-level concepts about the patient). Degree of obesity, for example, influences the overall level of intervention required, whilst physical ability need only be considered for physical activity recommendations. This patient information will, in turn, be derived by analysing data (e.g. the patient's height and weight) from one or more Monitoring process, and conversely, each Monitoring process will feed data into one or more Analysis process.

Although the example above relates to traditional patient-clinician encounters, a MADE loop for telemedicine systems can likewise be decomposed into a network of MADE sub-processes, such as shown schematically in Fig. 2(a), with the dashed borders delineating the high-level MADE processes and the solid lines the decomposed sub-processes. Wearable fall detection systems, for example, may involve multiple Monitoring processes by utilising data on a combination of body acceleration, body tilt and body vibration to detect falls¹⁸, and by considering multiple, independent telemedicine systems as one complex system of systems, it is clear that telemedicine systems may also encapsulate multiple Decision and Effectuation processes.

3.2.2. Serial decomposition

In addition to parallel decomposition, a MADE process may also be subject to serial decomposition, reflecting the fact that a process may comprise a sequence of sub-steps as shown in Fig. 2(b) for three Analysis processes (delineated by the dashed borders). In fact, in the symptom avoidance scenario presented in Section 3.1, the Monitoring process for measuring heart rate can be decomposed by considering the sub-steps required to realise its function. For example, it may comprise a first step of measuring the patient's ECG (i.e. the electrical activity of his or her heart) followed by a second step of deriving the patient's heart rate from the R-R peaks in the signal.

4. Application of the MADE Framework

4.1. Mapping clinical guidelines to MADE networks

In the MobiGuide project, we have used the MADE framework to analyse a clinical guideline for gestational diabetes mellitus (GDM)³ and derive from it a corresponding network of MADE processes¹⁹. Complementary to the knowledge acquisition methodology of Shalom et al. in 2008²⁰ for transforming a narrative guideline into a computer-interpretable guideline, this process of performing a MADE analysis was applied in the MobiGuide project to identify a set of distribution requirements for the GDM case.

To illustrate the process of performing a MADE analysis, we give a simplified case: the condition leading to the decision to increase the carbohydrates intake of the patient. The relevant extract from the 23-page GDM guideline is shown in Fig. 3. In consultation with expert clinicians we performed a manual MADE mark-up on it as indicated by underlining followed by inserted [M] for Monitoring, [A] for Analysis and [D] for Decision. Effectuation processes are not usually contained within the narrative guideline but are implied by the decisions to be made.

From the mark-up of this guideline, we can construct a corresponding MADE process model, which, for the text extract shown in Fig. 3, is best represented as comprising the following 6 processes (Fig. 4):

- A Monitoring process for measuring fasting urinary ketone levels.
- A Monitoring process for measuring carbohydrates intake. Note that this process is only implied in the excerpt in Fig. 3 but is described elsewhere in the guideline.
- An Analysis process for detecting ketonuria, i.e. the presence of three or more positive ketonuria measurements in one week.
- An Analysis process for detecting compliance to dietary prescription.
- A Decision process for deciding to increase carbohydrates intake based on the analysis of the patient's urinary ketone levels and carbohydrates intake.
- An Effectuation process for increasing the patient's carbohydrates intake.

‘‘... we routinely monitor fasting urinary ketones [M] in women with GDM. The patient measures ketonuria using urine strips [M]. Monitor ketonuria routinely means to measure ketones in the urine every day at fasting conditions [M]. ... The results of ketonuria could be: a) positive (++) ; b) positive (+) ; c) negative (+/-) ; d) negative (-) ; e) negative (--). ... In case of ketonuria detection (the number of ketonuria measurements with result ‘‘positive’’ is equal or higher than 3 in a period of time of one week) [A] :- If the patient was COMPLIANT with the prescribed diet [A], the nurse decides to increase the carbohydrates intake either at dinner or at bedtime: the amount of carbohydrates at dinner or at bedtime is increased by 1 unit (10 grams [D]). ...’’

Fig. 3. Extract from the MADE markup version of the GDM guideline³ leading to a decision to increase carbohydrates intake. The MADE mark-up is indicated by underlining followed by inserted [M] for Monitoring, [A] for Analysis and [D] for Decision.

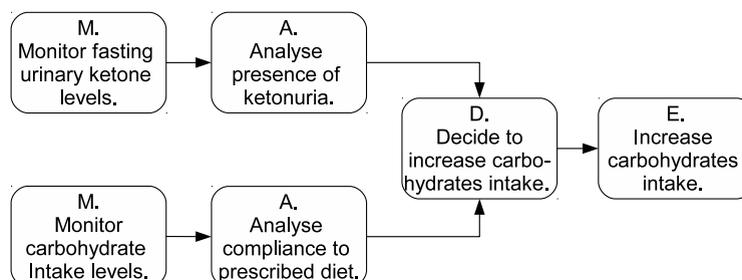


Fig. 4. Fragment of the MADE process model corresponding to the text extract from the GDM guideline shown in Fig. 3.

In Fig. 4, the arrows depict the flow of data from one process to another, showing, for example, that the decision to increase carbohydrates intake depends on the analysis of fasting urinary ketones and carbohydrates intake. In this case, the specific logical condition for triggering the decision is the conjunction of detection of ketonuria and compliance to dietary prescription, but for simplicity, this is not captured explicitly by the figure. Likewise, the figure only shows the required input and target output for the detection of ketonuria but not the logical condition governing the Analysis process, which is presence of three or more positive ketonuria measurements in one week.

Furthermore, although it is implied or explicitly stated in the text extract that the Monitoring, Decision and Effectuation processes are to be performed manually (by the patient for the Monitoring and Effectuation processes and by a nurse for the Decision process), it is nevertheless useful to capture these manual processes in the knowledge base. This allows the knowledge-based system to support such processes by, for example, providing regular reminders to the patient and validating the measured data. Indeed, if clinically validated as safe and feasible by the clinicians, some processes may also be completely delegated to the system. Thus as part of the knowledge acquisition methodology that is adopted in the MobiGuide project, the guideline developers (including the expert clinicians) would add explicit information about which parts of the process could be done by the MobiGuide system. In this example, the decision to increase carbohydrates intake was delegated completely to the system¹⁹. This delegation was done by the expert clinicians during a subsequent step in the guideline formalization process.

4.2. Design and implementation of a mobile MADE-based system

In conjunction with analysing the GDM guideline, we have also applied the MADE framework to derive an object-oriented prototype of a MADE-based decision support system that executes guidelines as a network of concurrent MADE processes²¹. In the MobiGuide project, the MADE-based system is implemented in Java on an Android smartphone, and it collaborates with a decision support system at the back-end to provide guidance to the patient. For example, whenever a patient’s treatment is started (or changed), a guideline “projection”, consisting of the currently relevant part of the relevant guideline, is built by the back-end decision support system and transmitted to the smartphone for the MADE-based decision support system. With the projected guideline, the MADE-based system can then operate autonomously, communicating with the other components on the Body Area Network to receive data

from, and output messages to, the patient. However, for simplicity, we abstract from these details here and regard the MADE-based system as interacting with a generic knowledge source, data source and effector component.

Like conventional knowledge-based systems, the MADE-based system consists of two main components as shown in Fig. 5: a Knowledge Base for storing the knowledge projected from a knowledge source and a Reasoning Component for applying the knowledge to patient data as it is received from sensors and other data sources. However, in our system, the Knowledge Base serves a less important role during run-time, as the projected knowledge is processed into and is embodied inside a network of MADE processes in the Reasoning Component. Whenever new knowledge is added to the Knowledge Base, it is parsed and interpreted immediately by the MADE Builder, resulting in the instantiation of multiple MADE objects, with each object corresponding to a particular process. For the text extract of the GDM guideline for example (Fig. 3), the Reasoning Component would, during run-time, contain six MADE objects reflecting the six MADE processes in Fig. 4. The MADE objects communicate with each other and with other components via a publish-subscribe mechanism, ensuring that the typology of the MADE network is completely configurable and depends solely on the input guideline.

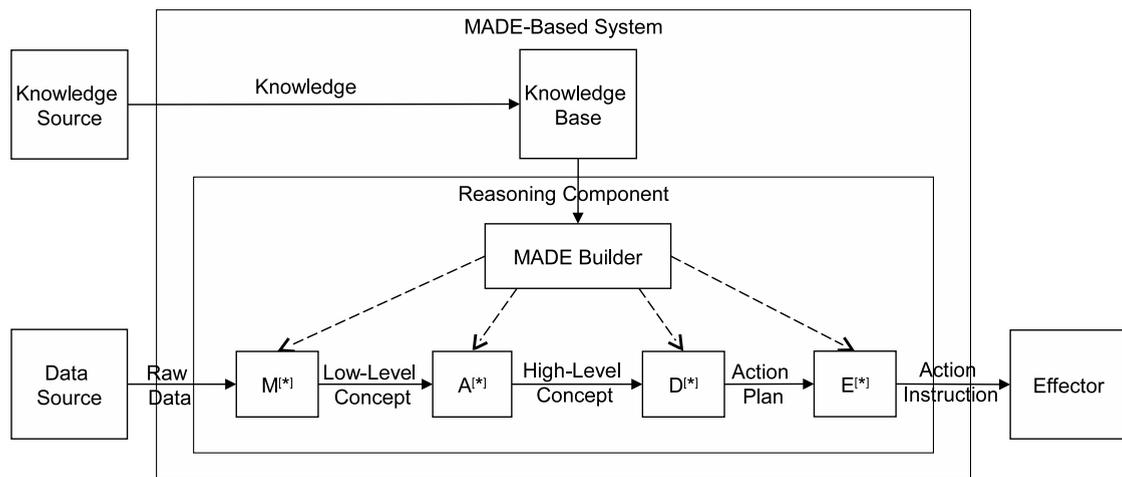


Fig. 5. The component architecture of the MADE-based system. The system contains many MADE objects in general as depicted by the $[\ast]$, and for simplicity, administrative and interfacing components are not illustrated.

The first prototype of the MADE-based system, which has been tested with the other components in a prototype MobiGuide system, can only perform some basic and specific knowledge processing and reasoning functions, namely to check the compliancy of a patient in measuring her blood glucose levels and to notify the patient and the back-end decision support system in case of non-compliance. As shown in Fig. 5, it was also assumed for the first prototype that the Analysis and Decision processes can all be performed automatically by the system and do not require interaction (e.g. confirmation) with the patient or caregiver. Nevertheless, we believe the research and implementation based on the MADE framework already completed demonstrates the feasibility of modelling and realising guideline-based reasoning as a network of concurrent processes.

Furthermore, although the MADE-based system was only implemented and tested on a single device, the results can be generalised to demonstrate the feasibility of using the framework to develop distributed and device-independent knowledge-based systems. Since the specific functionality of a MADE-based system depends only on the content of the knowledge base, distribution of MADE processes can be achieved by simply replicating the same MADE-based system across multiple devices and allocating different parts of the knowledge to each device, possibly with overlaps. Therefore, if it is assumed that an appropriate communications infrastructure is available, devices can be plugged into and out of the system as required and, by the appropriate re-distribution of knowledge, not affect the functionality of the overall system.

5. Discussion and Future Work

5.1. Mechanisms for the distribution of knowledge

Our experience demonstrates that the MADE framework can indeed support the development of distributed and device-independent knowledge-based systems right through to coding. Here we address the possible mechanisms for distributing knowledge, and thereby processes, across the different devices that constitute the system. One possible option, for example, is to have a centralised intelligent system that operates at the meta-level to determine the best distribution of knowledge, receiving as inputs the properties of the knowledge and of the available devices and outputting a corresponding mapping between each device and each partition of knowledge.

However, this approach may not be optimal for the dynamic distribution of knowledge, which may be desirable to account for the time-varying properties of each device (e.g. battery levels and available memory) and of each communications channel (e.g. available bandwidth). To achieve a high degree of dynamism would require a highly reliable communications channel between the centralised knowledge distributor and each device, which is inherently not available. Therefore, we plan to adopt a decentralised approach instead, with each device acting as an intelligent agent in a peer-to-peer network, capable of passing knowledge, and therefore functionality, to other devices depending on its current state.

Although it would offer much more dynamism to the distribution, one major challenge for the decentralised approach is the inherent difficulty of validating the behaviour resulting from the distribution. In the medical domain, safety and efficacy of a clinical intervention are of great importance, and they must not be jeopardised by any particular distribution of knowledge. Thus amongst the issues to solve in the future, a major one relates to verification of the correct semantics and hence correct behaviour of the overall distributed system, given all possible divisions of functionality together with the (well known) problems associated with concurrency. However, to realise any form of automated knowledge distribution, the MADE framework must first be extended into a full ontology, which is currently under development, for the complete specification of the MADE network, including the properties of each MADE process and of each device that govern the best distribution of knowledge, and therefore processes, in a telemedicine system.

5.2. Distributed reasoning in real-time and under uncertainty

Although the distribution of knowledge is intended to optimise overall system performance given the properties of the constituent devices, problems may nevertheless occur and must be pre-empted. For example, it is implicitly assumed that each MADE process can operate in real-time, but this may not be possible in some situations, such as when a MADE process requires complex data processing or when a device (e.g. a PC or smartphone) must perform other resource intensive tasks (e.g. video gaming) in addition to executing MADE processes.

To account for such situations, it may be useful to implement anytime algorithms in the MADE processes such that an output, albeit a possibly approximate one, will always be generated within the real-time constraints of the system. However, this implies that each MADE process must also be able to reason under uncertainty, as its inputs may now be of varying quality. Indeed, uncertainty is inherent in the system due to the noise in the sensing devices, and this may, in the worst-case scenario, cause the MADE processes to produce conflicting results. For example, one Decision process may, based on a particular set of patient information, recommend the patient to perform some exercise, whilst another Decision process, with another set of patient information, recommends the patient to refrain from exercising.

One possible approach to resolve such conflicts is to design additional mediating MADE processes into the system. Regarding conflicting exercise recommendations for example, a mediating Decision process may be used to decide which recommendation is clinically safer. Alternatively, the conflict resolution process may be embedded into the effectuation of the recommendations, or the two Decision processes may be combined into one complex Decision process such that only one of the two recommendations can be outputted at any given time. In any case, it is clear that the resulting MADE processes and MADE network will become more complex, and it is a future objective to develop the appropriate mechanisms for achieving increased robustness in a MADE-based system.

5.3. Decomposability of clinical knowledge

One of the main but implicit assumptions of the MADE framework is that the clinical knowledge can be partitioned and divided across different MADE processes. Although the validity of this assumption has been tested albeit informally in Section 4.1 for clinical guidelines, it is useful, at this point, to discuss the applicability of the framework to other forms of clinical knowledge.

For example, the Analysis process has been implicitly assumed to be a process of deduction, with the knowledge required for diagnosis expressed in the form of implications from lower-level concepts to higher-level concepts. As a result, the knowledge can be readily partitioned into a set of Analysis processes, with each implication corresponding to one single process. However, from an AI perspective, the process of diagnosis is one of abduction²², with the knowledge expressed in the form of implications from higher- to lower-level concepts²³. This implies that multiple implications need to be considered together when analysing data, as the process of diagnosis is now that of choosing an appropriate higher-level concept (e.g. common cold or influenza) to explain the observed lower-level concepts²² (e.g. symptoms of coughing, sneezing and sore throat²⁴). Thus in this case, partitioning the knowledge into a set of distinct Analysis processes would require a more complex procedure.

Similarly, the knowledge embodied in a Decision process is best represented in the form of an if-then-else statement, with the condition being the input from an Analysis process and the consequent being the plan. However, in the AI domain, planning is classically seen as the process of generating the appropriate sequence of actions given the goal state, the initial state and the set of all possible actions and their effect on the state of the world²⁵. Although it may still be possible, it is clear that under this model of the planning problem, there is no simple means of representing the knowledge as a set of Decision processes. The ability to partition the planning problem into a set of concurrent processes would seem to imply the ability to identify non-interacting sub-goals in the problem and to classify each action according to the specific sub-goal that it relates to.

6. Conclusion

Telemedicine systems are, by definition, distributed across multiple devices. However, unlike that of conventional distributed systems, the configuration of a personalised telemedicine system is only determined after design-time by considering, amongst other factors, the specific needs of the individual patient (e.g. which specific sensor(s) they need for their condition(s)). Therefore, to enable guideline knowledge to be partitioned and distributed at run-time across a distributed knowledge-based telemedicine system, we have developed and tested a conceptual framework for modelling and executing guidelines as networks of concurrent processes.

At the core of our framework is the MADE model of a disease management process, which comprise a network of Monitoring, Analysis, Decision and Effectuation processes controlling the state of the patient and his or her surroundings. The framework was found to be applicable through the full development trajectory, from the analysis phase of modelling clinical guidelines to the design and implementation phases of developing a complete MADE-based system. Indeed, a first prototype of such a system was demonstrated as part of the MobiGuide system, which is scheduled to be piloted on two patient groups in Spain and Italy.

Although focused on the telemedicine domain, the MADE framework may also be equally applicable to other application domains which involve distributed knowledge-based systems. In the robotics domain for example, knowledge may also need to be shared and distributed across multiple robots such that they can collectively and autonomously solve a complex task. However, this currently requires the knowledge of the domain to be expressible in the form of guidelines, and it remains an open question whether the MADE framework can be generalised for other knowledge representation formalisms as well.

In the future, the MADE framework will be extended by considering the different determinants on the best distribution of knowledge across multiple devices. This will result in the development of a full ontology for the MADE processes, which will, in turn, be used to finalise the design of the MADE-based system to support, for example, anytime reasoning and uncertain reasoning. At the meta-level, this ontology will also form the basis for developing an intelligent mechanism to distribute knowledge automatically across multiple devices in the context of a distributed decision support system.

Acknowledgements

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. Jones VM, Bults RGA, Konstantas DM, Vierhout PAM. Healthcare PANs: Personal Area Networks for trauma care and home care. Presented at Fourth International Symposium on Wireless Personal Multimedia Communications (WPMC); 2001.
2. Camm AJ, Lip GY, De Caterina R, Savelieva I, Atar D, Hohnloser SH, 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 Journal* 2012;**33**:2719–2747.
3. Rigla M, Tirado R, Caixàs A, Pons B, Costa J. *Gestational Diabetes Guideline CSPT*. MobiGuide Project (FP7-287811); 2013. Version 1.0, 12/02/2013. Internal document.
4. van Melle W. A domain-independent production-rule system for consultation programs. In: *Proceedings of the 6th international joint conference on Artificial intelligence (IJCAI'79)*; vol. 2. 1979, p. 923–925.
5. Peleg M. Computer-interpretable clinical guidelines: A methodological review. *Journal of Biomedical Informatics* 2013;**46**:744–763.
6. Peleg M, Tu S, Bury J, Ciccarese P, Fox J, Greenes RA, et al. Comparing Computer-interpretable Guideline Models: A Case-study Approach. *Journal of the American Medical Informatics Association* 2003;**10**:52–68.
7. Sutton DR, Fox J. The Syntax and Semantics of the PROforma Guideline Modeling Language. *Journal of the American Medical Informatics Association* 2003;**10**:433–443.
8. Carson ER, Cramp DG, Morgan A, Roudsari AV. Clinical Decision Support, Systems Methodology, and Telemedicine: Their Role in the Management of Chronic Disease. *IEEE Transactions on Information Technology in Biomedicine* 1998;**2**:80–88.
9. Lasierra N, Alesanco A, Guillén S, García J. A three stage ontology-driven solution to provide personalized care to chronic patients at home. *Journal of Biomedical Informatics* 2013;**46**:516–529.
10. Kephart JO, Chess DM. The Vision of Autonomic Computing. *Computer* 2003;**36**:41–50.
11. OpenClinical. <http://www.openclinical.org/>; (Accessed May 28, 2014).
12. Lasierra N, Alesanco A, García J, O'Sullivan D. Data Management in Home Scenarios Using an Autonomic Ontology-Based Approach. In: *2012 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*. 2012, p. 94–99.
13. Babcock B, Babu S, Datar M, Motwani R, Widom J. Models and Issues in Data Stream Systems. In: *Proceedings of the Twente-first ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*. 2002, p. 1–16.
14. Broens T, Bults R, Fung N, Jones V, Larburu Rubio N, Widya I, et al. *Deliverable 2.3: Distribution of Decision Support Functionality*; 2012. Available at <http://www.mobiguide-project.eu/downloads/category/10-public-deliverables> (Accessed March 31, 2014).
15. Seyfang A, Miksch S, Marcos M. Combining diagnosis and treatment using ASBRU. *International Journal of Medical Informatics* 2002;**68**:49–57.
16. Russell S, Norvig P. *Artificial Intelligence: A Modern Approach*. Pearson Education, Inc.; 3 ed.; 2010.
17. NICE. *Obesity: Guidance on the prevention, identification, assessment and management of overweight and obesity in adults and children*. National Institute for Health and Clinical Excellence (NICE), UK; 2006.
18. Yu X. Approaches and Principles of Fall Detection for Elderly and Patient. In: *2008 10th IEEE International Conference on e-Health Networking, Applications and Services*. 2008, p. 42–47.
19. Shalom E, Quaglini S, Sacchi L, Panzarasa S, Tormene P, Napolitano C, et al. *Deliverable 5.2.1: Design Progress Report*. MobiGuide Project (FP7-287811); 2013. Internal document.
20. Shalom E, Shahar Y, Taieb-Maimon M, Bar G, Yarkoni A, Young O, et al. A quantitative assessment of a methodology for collaborative specification and evaluation of clinical guidelines. *Journal of Biomedical Informatics* 2008;**41**:889–903.
21. Shalom E, Quaglini S, Sacchi L, Panzarasa S, Tormene P, Parimbelli E, et al. *Deliverable 5.2.2: WP5 Development Progress Report*. MobiGuide Project (FP7-287811); 2013. Internal document.
22. Douven I. Abduction. In: Zalta EN, editor. *The Stanford Encyclopedia of Philosophy*. Spring 2011 ed.; 2011, .
23. Brachman R, Levesque H. *Knowledge Representation and Reasoning*. Morgan Kaufmann; 2004.
24. *Cold or flu?* National Health Service (NHS), UK; 2012. Available at <http://www.nhs.uk/Livewell/coldsandflu/Pages/Isitacoldorflu.aspx> (Accessed March 31, 2014).
25. Ghallab M, Nau D, Traverso P. *Automated Planning: Theory & Practice*. Morgan Kaufmann; 2004.