Using social integrity constraints for on-the-fly compliance verification of medical protocols

Anna Ciampolini, Paola Mello, Marco Montali
DEIS, University of Bologna, Viale Risorgimento 2, 40136 Bologna, Italy
{aciampolini, pmello, mmontali}@deis.unibo.it

Sergio Storari
ENDIF, University of Ferrara, Via Saragat 1, 44100 Ferrara, Italy
sstorari@ing.unife.it

Abstract

We propose to adopt a formalism, based on social integrity constraints (ICs), for specifying social interactions between actors involved in a guideline. ICs allow us to represent interaction protocols using a logic formalism and to perform an on-the-fly verification of the protocol's application compliance, based on an abductive proof procedure which operates on relevant events occurred during its application. The paper presents the results of a first trial performed on a microbiological clinical guideline which exploits the potentialities of the formalism in representing and verifying the compliance to medical guidelines.

1 Introduction

Medical guidelines [3] are clinical behaviour's recommendations used to help and support physicians in the definition of the most appropriate diagnosis and/or therapy within determinate clinical circumstances.

Unfortunately, guidelines are today described using several formats, such as flow charts and tables, so that physicians are not properly supported in the detection of possible errors and incompletenesses: it is difficult to evaluate who made an error within the protocol's flow and when. As a consequence, guideline's application often looses its benefits.

In this work we make use of a formal language for the specification of agents' interaction protocol in order to describe medical guidelines. This formalism, developed within the SOCS European project [5] with our collaboration, is based on social integrity constraints (ICs) [1]. ICs allow us both to represent interaction protocols using a logic formalism and to perform an on-the-fly verification of the protocol's application compliance, based on an abductive proof procedure which operates on relevant events occurred during its application.

We show that such a formalism is general enough to allow us to describe medical protocols. The main advantage of using ICs is the capability of discovering some forms of inconsistency and to perform an on-the-fly verification of the protocol's application on a specific clinical event.

In order to effectively test the potentialities of this approach, we formalized, making use of experience acquired in a previous project [4], a microbiological guideline which describes how to manage an infected patient from his/her arrival at a hospital's emergency room to his/her recovery. We then tested this guideline on a set of clinical trials.

Proceedings of the 18th IEEE Symposium on Computer-Based Medical Systems (CBMS'05)
1063-7125/05 $20.00 © 2005 **IEEE**

2 The SOCS formalism

The SOCS European project [5] gives a formal definition based on computational logic both of agents and of societies of interacting agents in a global computing setting. SOCS social infrastructure (SOCS-SI) consists of a social model and an abductive proof procedure for the on-the-fly verification of agent's compliance to protocols.

A protocol is specified by a set of rules (called integrity constraints, ICs) which controls events' flow within society. In the SOCS framework, an observable event is represented as an atom $\mathbf{H}(Event, T)$, named *happened* event. Given a set of *happened* events (at certain time), ICs express which new *expectations* the society has about agents' "ideal" behavior in order to comply with the considered protocol. An *expectation* may be positive (represented with symbol $\mathbf{E}(Event, T)$, named *expected* event), meaning that the society expects that $Event$ happens (at time T), or negative (represented with symbol $\mathbf{NE}(Event, T)$, named *not expected* event), meaning that the society expects $Event$ not to happen.

Integrity constraints' formalism allows us to consider conjunction and disjunction of *happened* events and *expectations* as well as to specify temporal constraints over them. The abductive proof procedure acquires *happened* events and consequently generates corresponding *expectations*. Each *expectation* can be fulfilled or not by the events occurred in the society; then if an agent does not behave as expected, the proof procedure detects run-time protocol's violations.

3 Using the SOCS formalism to specify and verify a medical guideline

In order to experiment the potentialities of the SOCS formalism, we have considered a fragment of a clinical guideline applied in hospitals for microbiological infections' treatment. The guideline may be structured in seven macro phases, from patient's arrival at the hospital's emergency room with infection symptoms to his/her recovery after a suitable infection's therapy.

In order to formalize this guideline, we detected, first of all, all the actors involved (e.g. patient, ward's physician, microbiological laboratory, etc.) and secondly pointed out all the actions which should be executed (or not, i.e. expected or not expected) for an appropriate patient's disease treatment. Each actor has been then mapped into an agent with a specific role, and actors' interaction has been modeled as a set of ICs. For example, the following IC

$$\begin{array}{l} \mathbf{H}\left(enter\left(Patient, emergency\_ward\right), Tent\right) \\ \rightarrow \mathbf{E}\left(visit\left(Physician, Patient\right), Texam\right) \\ \wedge Texam < Tent + 6 * 60 \end{array} \tag{1}$$

expresses that when a patient arrives at the emergency room (at time $Tent$), we expect that at least one physician would visit him (at time $Texam$) within the deadline of 6 hours. This deadline is expressed as a CLP constraint, which says that $Texam$ should be lower than $Tent$ plus 6 hours.

The complete specification of this protocol consists of about 20 social ICs. It has been tested via the SOCS-SI software (a JAVA-Prolog-CHR implementation of the proof procedure), using different set of events, compliant and not. For instance, a non compliant set is the following: a patient ($patientA$) arrives at the hospital's emergency room at time 10, but no physician visits him within 6 hours. The event $\mathbf{H}\left(enter\left(patientA, emergency\_ward\right), 10\right)$ matches with the antecedent of (1), generating the expectation in the consequent that a physician should visit $patientA$ at time

$Texam$, such that $Texam \leq 10 + 6 * 60$. No event is afterward registered until this deadline, therefore a violation is raised by the proof procedure.

Note that SOCS-SI can work indifferently off-line (i.e. giving the complete list of happened events) and on-line (i.e. giving the events at real-time). Consequently, in order to use effectively SOCS-SI for guidelines' on-the-fly monitoring we need to extract "medical" events at real-time. Usually, these events can be observed by monitoring database informations' changing; thus, we are working on a software module that controls database's changes (such as insertion or modification of records) related to a guideline and convert them in events that can be processed by SOCS-SI. This module will permit the integration of SOCS-SI with almost all existing medical software systems without requiring changing on them.

4 Conclusions and future works

We have shown how a simple medical guideline may be mapped into a set of social integrity constraints in the context of SOCS' infrastructure, to the purpose of enabling an on-the-fly verification about the compliance of the hospital staff to it. We have also successfully tested this specification using the SOCS-SI software with some set of events, compliant and not. Of course, this is only the first step towards an effective tool for defining and verifying guidelines in a clinical environment. In literature, several formalisms have been proposed for representing medical protocols, like for example GLARE [6] and PROforma [2]. These are complete tool capable to manage both guidelines acquisition and execution, but, as for as we are concerned, their are not able to verify compliance of actions and interactions of the kind here presented.

Therefore, we are extending SOCS-SI for the clinical environment by adding new features such as: a graphical language for guideline's description (and a translator into ICs) and a module that generates guideline's related events by monitoring hospital's databases.

5 Acknowledgements

This work was partially funded by the IST programme of the EU under the IST-2001-32530 SOCS project, by MIUR under the COFIN2004 project "MASSIVE: Sviluppo e verifica di sistemi multiagente basati sulla logica" and by "SPINNER: Servizi per la Promozione dell'INNovazione e della Ricerca" Project 45/04. The authors would like to thank Luca Porreca for his help.

References

[1] M. Alberti, M. Gavanelli, E. Lamma, P. Mello, and P. Torroni. Specification and verification of agent interactions using social integrity constraints. *Electronic Notes in Theoretical Computer Science*, 85(2), 2003.

[2] J. Fox, N. Johns, A. Rahmanzadeh, and R. Thomson. Disseminating medical knowledge: the proforma approach. *Artificial Intelligence in Medicine*, 14:157–181, 1998.

[3] C. Gordon. Practice guidelines and healthcare telematics; towards an alliance. *Health telematics for clinical guidelines and protocols*, pages 3–15, 1995.

[4] E. Lamma, P. Mello, G. Modestino, A. Nanetti, F. Riguzzi, and S. Storari. An intelligent medical system for microbiological data validation and nosocomial infection surveillance. *Proc. of the International Symposium on Computer Based Medical Systems (CBMS) 2002*, pages 13–20, 2002.

[5] Societies Of ComputeeS (SOCS), IST-2001-32530, http://lia.deis.unibo.it/research/socs/ .

[6] P. Terenziani, P. Raviola, O. Bruschi, M. Torchio, M. Marzuoli, and G. Molino. Representing knowledge levels in clinical guidelines. *Proceedings of the Join European Conference on Artificial Intelligence in Medicine and Medical Decision Making*, 1620:254–258, 1999.