MEDINFO 2007
HISA Ltd Australia
© 2007 The authors. All rights reserved.

Applying AI Model-Checking Techniques to Clinical Guidelines

Laura Giordan[a], Alessio Bottrighi[a], Stefania Montani[a], Paolo Terenziani[a]

[a]*DI, Univ. del Piemonte Orientale "Amedeo Avogadro", Spalto Marengo 33, 15100 Alessandria, Italy,*
E-mail: {terenz, alessio.bottrighi, laura, stefania}@mfn.unipmn.it

Abstract

We describe the advantages of extending computer-based approaches to clinical guidelines with the facilities provided by model-checkers developed by the AI community.

Introduction and background

Clinical guidelines represent the current understanding of the best clinical practice, and constitute an important area of research in Artificial Intelligence (AI) in medicine. Several computer-based approaches have been introduced to automatically manage clinical guidelines. GLARE (Guideline Acquisition, Representation and Execution) is a prototypical system which has been built from 1997 under the leadership of physicians of Az. Osp. S. Giovanni Battista, Turin, Italy [1]. It is a domain-independent system to acquire, represent and execute clinical guidelines, which has been tested on several guidelines, including asthma, cardiac diseases, ischemic stroke.

Recently, in AI, the *model-checking* techniques have gained an important role. They are widely used, e.g., for agent verification. In this work, we extend GLARE to explore the advantages of applying such techniques to clinical guidelines.

Methods

In order to enhance GLARE with model-checking facilities, we provided a loosely-coupled integration of GLARE with the model-checker SPIN [2]. Roughly speaking, we have devised a tool to map clinical guidelines acquired by the GLARE system into the Promela language, which is the language used by SPIN. Promela allows a high level model of a distributed system to be defined in an extended pseudo-C code, including synchronization primitives and message exchange primitives. Once we have the translation of GLARE's guidelines in Promela code, we can use SPIN as a general-purpose engine to prove any property that can be expressed in the temporal logic LTL. In fact, SPIN automatically translates each process defined in the Promela code into a finite automaton representing the global state space of the system, and the global behaviour of the system is obtained by computing an asynchronous interleaving product of automata. The property which has to be verified on the system is passed to the verifier through an interface, which maps it into a temporal formula, as required by SPIN. SPIN automatically converts the negation of the temporal formula into an automaton and computes its synchronous product with the system global state space. If the language of the resulting automaton is empty then the property is true on all the possible execution of the system (otherwise the verifier provides a counterexample for the property).

Results

The methodology above provides a way of providing GLARE with the possibility of automatically checking several types of properties concerning guidelines. Specifically, we have identified four different classes of properties:

i) **Properties concerning a guideline "per se"**. One can check if the guideline contains a path of actions satisfying a given set of conditions (e.g., a path including actions X, Y and Z, or a path nor requiring a given laboratory test, or a path requiring only a given set of resources, and so on);

ii) **Properties of a guideline in a given context.** Specific contexts may impose limitations on executable actions, related, e.g., to the lack of resources (e.g., laboratory instruments). The consequences of such limitations may be automatically checked using the model checker. For instance, the model checker can prove whether there is or not a therapy for a patient affected by a given disease, in the case a specific set of resources is available (not available).

iii) **Properties of a guideline when applied to a specific patient.** Provided that the model checker has in input all the data in the patient record, the feasibility of a given action or path of actions on the patient can be proved.

iv) **Integrated proofs.** Since SPIN is flexible and task-independent, any combination of the above types of proofs is feasible. E.g., one may ask whether, given a patient with a specific disease and set of symptoms, and given a specific set of resources, there is a path in the guideline which applies to the patient and satisfies a given set of properties.

Acknowledgements

This work has been partially supported by Koine Sistemi and by MIUR PRIN 2005 "Specification and verification of agent interaction protocols".

References

[1] P. Terenziani, G. Molino, M. Torchio. A Modular Approach for Representing and Executing Clinical Guidelines. *Artif. Intelligence in Medicine 23* (2001) 249-276.

[2] G.J.Holzmann, The SPIN Model Checker. Primer and Reference Manual. Addison-Wesley, 2003