META-GLARE: a shell for CIG systems

Alessio BOTTRIGHI, Stefania RUBRICHI, and Paolo TERENZIANI

Computer Science Institute, DISIT, Univ. Piemonte Orientale, Alessandria, Italy

Abstract. In the last twenty years, many different approaches to deal with Computer-Interpretable clinical Guidelines (CIGs) have been developed, each one proposing its own representation formalism (mostly based on the Task-Network Model) and execution engine. We propose META-GLARE, a shell for easily defining new CIG systems. Using META-GLARE, CIG system designers can easily define their own systems (basically by defining their representation language), with a minimal programming effort. META-GLARE is thus a flexible and powerful vehicle for research about CIGs, since it supports easy and fast prototyping of new CIG systems.

Keywords. Medical Informatics Applications [L01.700.508], Decision Making, Computer Assisted [L01.700.508.100]

Introduction

Clinical practice guidelines (CPGs) play an important role in medical practice, and computerized support to CPGs is now one of the most central areas of research in Artificial Intelligence in medicine. In recent years, many groups have developed different computer-assisted management systems of Computer Interpretable Guidelines (CIGs) (see, e.g., [1, 2]). Though each approach proposes its own representation formalism, most approaches model CPGs as a Task-Network Model (TNM): a hierarchical model of the guideline control flow as a network (graph) of specific tasks (represented by nodes). From the architecture point of view, most CIG systems provide specific support for at least two subtasks: (i) CIG acquisition and representation and (ii) CIG execution.

However, there are also important diverging features between the different CIG systems, also due to the fact that many of such systems are mostly research tools that evolve and expand to cover an increasing number of phenomena/tasks. This is, for instance, the case of GLARE (Guideline Acquisition, Representation, and Execution), the prototypical system we have been building since 1996 in cooperation with ASU San Giovanni Battista in Turin, one of the major hospitals in Italy [3]. The continuous need for extendibility lead us to a new and innovative idea: instead of building a new version of a CIG system whenever a new extension (typically to the representation language) is needed, we decided to work-out META-GLARE, a shell to easily build/extend CIG systems, minimizing the effort of system designers and developers.
1. Methods

Basically, META-GLARE takes in input a CIG formalism (based on the TNM model), and provides as output a CIG system for acquiring, consulting and executing CIGs expressed through the input formalism. The methodology we follow is

- We defined an open library of elementary components (e.g., textual attribute, Boolean condition, Score-based decision), each of which was equipped with methods for acquiring, consulting and executing it
- We provide system-designers with an easy way of aggregating such components to define node and arc types (constituting the representation formalism of a new system)
- We devised general tools for the acquisition, consultation and execution of CIGs, represented by TNM, which are parametric over the node and arc types (i.e., the definition of node and arc types are an input for such tools).

2. Results

In Figure 1, we show a simplified version of the architecture of META-GLARE. Oval nodes represent data structures, and rectangles represent computational modules. The DEFINITION_EDITOR module supports system-designers in the definition of a new system. It consists in four sub-components, to cope with the definition of (i) attribute types, (ii) node/arc types, (iii) graph constraints, and (iv) CIG formalism (where a CIG formalism is just a set of node/arc types and –possibly- of graph constraints). The output is an XML document in the CIG_FORMALISM_LIBRARY. Globally, the DEFINITION_EDITOR module manages the definition of the formalism of a new CIG system. The HDG_INTERPRETER deals with the aspects which common to all the systems that can be generated by META-GLARE. It consists of three sub-components: HDG_ACQUISITION, HDG_CONSULTATION, and HDG_EXECUTION. META-GLARE and its modules are developed as Java Applets (thus methods in Figure 1 are implemented by Java classes).

Figure 1. The (simplified) architecture of META-GLARE.
3. Discussion

META-GLARE provides a flexible tool for researchers in the area of CIGs, supporting easy and fast prototyping of new CIG systems. Using META-GLARE:

- the definition of a new system (based on a new representation formalism) is easy and quick;
- the extension of an existing system (through the modification of the representation formalism) is easy and quick.

In both cases, fast prototyping of the new (or extended) system is achieved. Programming is not needed at all if the new (or extended) CIG system has a new formalism, but uses attribute types (and types of constraints) already in the Library. Programming (by the developer of the new CIG system) is required only in case new attribute types (or new constraints) have to be added to the Library. In such a case, the methods for acquiring, consulting and executing attributes of the new types have to be programmed by the system developer. Notably, such programming is completely "local" and "modular": the new methods have to be programmed "in isolation" (without having to care of the rest of the system). No modification to any software component in METAGLARE architecture (see Figure 1) is required to integrate the new methods: META-GLARE automatically evokes them when needed for acquisition, consultation and execution. We thus look META-GLARE as a valuable vehicle to address new CIG phenomena, minimizing the software efforts needed for extending the CIG system (or for defining a new system). For instance, we are planning to use it

- To extend GLARE formalism with new features (new attributes in the description of nodes), needed to cope with comorbidities, and
- To implement a new system, geared towards education.

The implementation of META-GLARE is still ongoing: the acquisition engine is almost complete, while the execution component is under development.

Acknowledgements

The research reported in this paper has been partially supported by Compagnia di San Paolo, within the GINSENG project.

References