Distributed AI Technologies for Patient Management

Alberto Riva, Riccardo Bellazzi and Mario Stefanelli
Dipartimento di Informatica e Sistemistica
Università di Pavia, Italy

1 Introduction

Recent advances in information technology offer the chance to overcome some limitations that usually impair the widespread use of Artificial Intelligence in Medicine (AIM) methodologies. Typically, applications were designed as stand-alone systems implemented on dedicated hardware and software platforms. This design philosophy, that was motivated by the state of technology, led to a number of well-known problems. First, systems built for different applications were also completely different from the point of view of their usage, their interaction interfaces, their knowledge and data representation formalisms and, often, the hardware platforms they required. Second, they were not well accepted by end-users because the results of their reasoning processes appeared to focus on specific aspects of the problem at hand instead of integrating with the daily clinical practice and the overall management of the patient. Finally, the stand-alone and independent nature of the systems caused several problems, both at the practical level (installation, maintenance and upgrading of AI systems) and at the conceptual one (e.g., knowledge sharing and reuse in knowledge based systems) [6].

We believe that recent developments in information technology can support a more efficient and productive use of earlier and modern AIM methodologies. In particular, we will discuss how the use of network technologies, distributed architectures and multi-media capabilities enables a better integration of AIM tools inside a distributed medical information system. The advantages that we expect lie therefore in the following categories:

- *Availability*: distributed architectures enable users to easily access remote hardware and software resources. In this way, the same tools are available to a larger number of users, and are more easily maintained and managed by service providers.
- *Usability*: instead of developing an *ad-hoc* user interface for every different service, developers of distributed systems are forced to make them accessible through common network communication tools. From the point of view of the users, this means that all the different services available on the network can be used by means of a similar interface.
- *Acceptability*: the availability of multiple information servers on the same network makes it easier to integrate them inside a "virtual" medical workstation; in this framework, they are seen as commodities that, possibly together with other problem solvers based on non-AI methodologies, can be used to perform decision making in a real-world medical context [3].

On the other hand, the design of the overall information system may be enhanced through the use of appropriate AI techniques. For example, multiple-agent architectures provide a sound theoretical basis for the design of distributed systems [4]. As a real example, in the next section we will describe a system for diabetic patients management designed to be used in a networked distributed environment and whose decision-making process is based on AIM techniques.

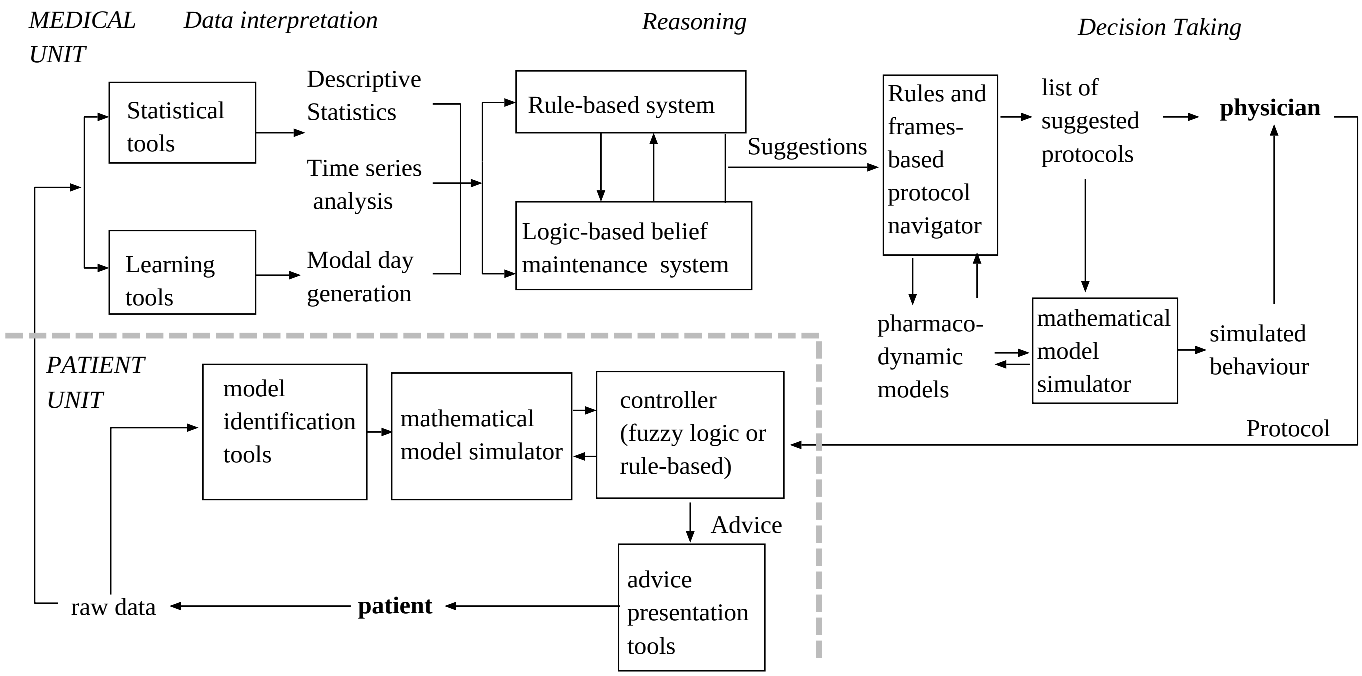

Figure 1: The architecture of the system.

2 System functionalities

2.1 Overview of the system

We are currently working on the design, implementation and testing of an intelligent telemedicine service to assist the management of Insulin Dependent Diabetes Mellitus (IDDM) patients [1]. The architecture of the system, designed to run in a telemedicine context, is based on a patient unit (PU) and a medical workstation (MW).

The PU provides assistance to the patient under the form of a set of local consultation procedures, autonomous decision-support tools and teleconsultation to the remote medical workstation. The MW deals with the long-term management of the patient by assisting the physician in choosing a suitable treatment protocol. To this purpose, it exploits both medical knowledge and clinical information, from both patients' home monitoring and the periodic evaluation of their metabolic control performed by the physicians.

The tools and the functionalities provided by the MW and the PU have been previously described in [7, 2]. An overview of the system, showing the methodologies employed in the different components, is presented in Figure 1.

The overall architecture implements a two-level hierarchical control scheme. The goal of the low level controller (implemented within the PU) is to provide *advice* on possible modifications to a predefined *therapeutic protocol*, based on a suitable *control law* applied to patient's data. The goal of the high level controller (implemented within the MW) is to select the appropriate *therapeutic protocol*, consisting of the insulin administration plan, the diet, general suggestions on the patient's lifestyle and the control law to be used by the low level controller.

The reasoning tools exploited by the two levels include probabilistic reasoning, rule based systems and mathematical modeling at the high level, and fuzzy, rule-based and traditional control algorithms at the low level.

In the rest of this paper we will show in greater detail how we exploited emerging information technologies to integrate the above mentioned tools in a system endowed with improved availability, usability and acceptability features.

2.2 System implementation and usage

We chose to base the interaction with the system on the HTTP protocol. This choice brings about a number of advantages: HTTP is a standard protocol and the number of available HTTP clients is growing extremely fast; the HTTP protocol is also very simple to implement and manage. Moreover, HTML, the usual language for information presentation within the HTTP protocol, provides good facilities for the low-cost creation of structured multimedia documents, endowed with graphical and user input handling capabilities. This means that a system based on HTTP and HTML is able to produce high-quality hypertextual output that can be used as a graphical user interface.

In order to exploit these features, we developed an HTTP server written in Common Lisp. The server is able to receive requests in HTTP form, to execute one or more Lisp functions according to the request received, and to present the output of the execution as HTML code. Applications can be loaded inside the server, which can be configured to invoke them when it receives an appropriate request. A system loaded in the HTTP server becomes available at once to every user who has access to the HTTP protocol; the users may interact with different applications using the same common interface and, finally, this framework makes it possible to seamlessly integrate AIM methodologies with other problem solvers, independently of their physical location.

HTTP and HTML can also be used to manage the communication with the PU. The PU design must take into account several additional features, related to the peculiarity of the patient/physician connection. The necessity of preserving information security and privacy, and the requirement of using low-cost public telephone networks led us to develop an extended version of HTTP, called STSP (Server To Server Protocol). STSP improves the reliability and the privacy of the communication, and allows us to implement special–purpose functions, such as data-base access and therapeutic protocol transmission management.

The PU interface is hence realized in HTML and uses an internal HTTP server to activate dedicated communication procedures to exchange data with the MW server. The MW can also serve as a *fire-wall* for the PU, providing the patient with a number of services available on the Internet and preserving the confidentiality of patient/physician communication.

We will now show how our system uses the capabilities of the HTTP server through an example of a typical MW task execution.

In addition to the automated reasoning tools used to suggest an appropriate protocol, the MW provides facilities to display and edit a selected protocol, to simulate it and to judge its adequacy.

Protocols are stored in the system as data structures and are organized in a taxonomy according to their clinical and therapeutic characteristics. When the user selects a protocol, the Common Lisp HTTP server is able to translate it into HTML, as shown in Figure 2. The protocol is represented as a table describing insulin types and doses to be taken at different day times. The graph plots the insulin activity over 24 hours calculated through a pharmacodynamic model [5], exploiting HTML's ability to embed images in text pages.

The user is then presented with a *fill-in* form (Figure 3) through which he can adjust all the components of the selected protocol. The standard HTTP method to submit the contents of a form is used to send the modified protocol back to the server, that updates its internal data structures accordingly by creating a new instance in the protocol taxonomy. Figure 4 shows a part of the protocol taxonomy using a knowledge-base browser embedded in the Common Lisp server. The new protocol can now be passed to a module (implemented using the MATLAB environment) that deals with the simulation of the patient's response using a suitable physiological model [5]. Finally, the results of the simulation, converted into graphical form, are sent back to the user and are presented as shown in Figure 5.

It is important to note that in the course of this process, the user was able to access the dif-

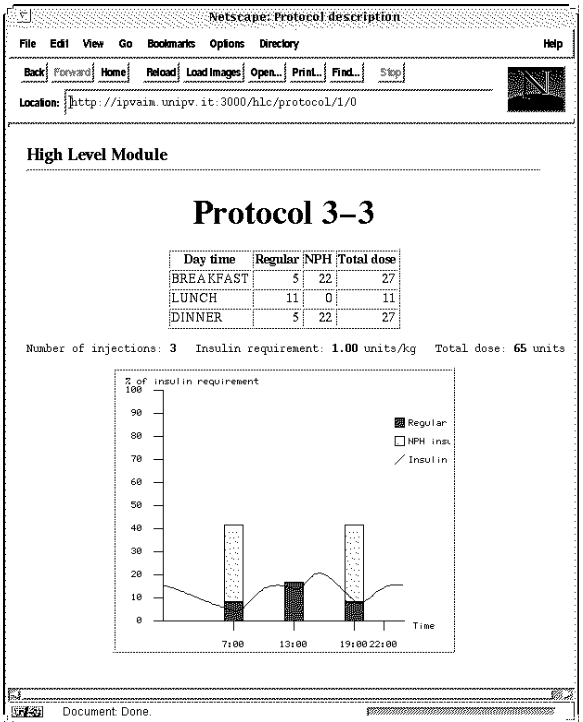

Figure 2: The representation of a typical insulin protocol.

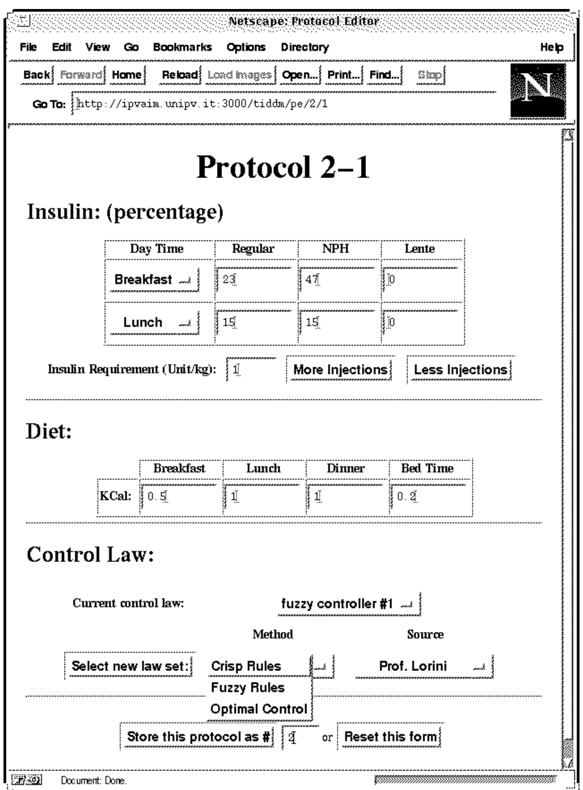

Figure 3: The protocol editor.

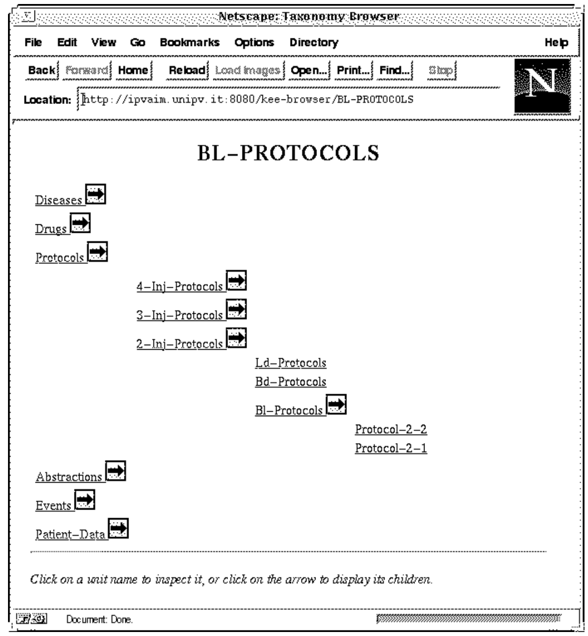

Figure 4: The protocol taxonomy as displayed by the HTML knowledge base browser.

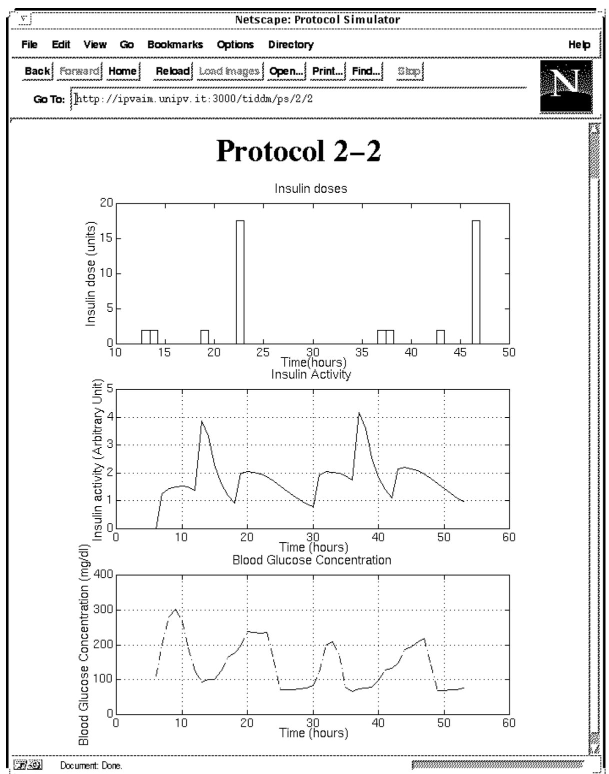

Figure 5: The simulation of glycemia and insulin dynamics.

ferent services offered by the system (e.g. protocol representation, simulation tools, graphical output capabilities) using only a common WWW browser, without the need to be familiar with a specialized interface or to have any knowledge of the inner working of the system. The same applies to the whole reasoning process of the MW, that integrates different formalisms (rules, probabilistic reasoning, physiological modeling) to provide support to both physicians and patients in their decision making activity.

3 Conclusions

In our view, the architectural and technological solutions presented above offer a way to improve the availability, usability and acceptability of current AIM methodologies. The use of the HTTP communication protocol improves the system availability, integrating it with the WWW infrastructure. On the other hand, we are currently working to extend the HTTP protocol to support alternative forms of interaction with the system. For example, in our application this is required by the nature of the connection between the PU and the MW, during which a bi-directional exchange of confidential patient information must take place.

By forcing all applications to adopt the HTML language for output presentation, we greatly enhance the usability of the system, since all interactions are based on a uniform and widely accepted standard. This advantage must of course be weighed against the limitations imposed by the language, in particular concerning user input management.

The application presented in our example is the result of the cooperation between several independent knowledge-based tools, each of which must be developed and maintained by a different set of experts. Our proposed framework could also be used to integrate AIM systems in a wider, multi-agent context. In this view, HTTP-based communication protocols would be used to support the cooperation and negotiation activities typical of autonomous agents, while HTML interfaces would provide a uniform view of the overall state of the agent community.

We are now continuing the development of the prototype of our system in order to be able to test it in a real clinical setting in the near future. This research will be carried on within the T-IDDM [1] project of the EEC "Telematics Applications" program.

References

[1] R. Bellazzi, C. Cobelli, E. Gomez and M. Stefanelli, The T-IDDM project: Telematic management of Insulin Dependent Diabetes Mellitus in: *Proceedings of Health Telematics '95*, 1995, M. Bracale, F. Denoth eds., Ischia, 271-276.

[2] R. Bellazzi, C. Siviero, M. Stefanelli, G. De Nicolao Adaptive controllers for intelligent monitoring Artificial Intelligence in Medicine, 7 (1995) 515-540.

[3] A. Farquhar, R. Fikes, W. Pratt, and J. Rice. Collaborative ontology construction for information integration. Technical Report KSL-95-63, Knowledge Systems Laboratory, Department of Computer Science, University of Stanford, Agosto 1995.

[4] M.P. Georgeff, A.S. Rao, The Semantics of Intention Maintenance for Rational Agents. Proceedings of IJCAI-95, pages 704–710, Montreal, Quebec, Canada, 1995.

[5] E. D. Lehmann and T. Deutsch. A physiological model of glucose-insulin interaction in type 1 diabetes mellitus. *Journal of Biomedical Engineering*, (14):235–242, 1992.

[6] M.A. Musen, Dimensions of Knowledge Sharing and Reuse. Computers and Biomedical Research, 1992, 25:435-467.

[7] A. Riva, R. Bellazzi, High Level Control Strategies for Diabetes Therapy. Lecture Notes in Artificial Intelligence (P. Barahona, M. Stefanelli, J. Wyatt eds.), 934, pages 185–196, Springer-Verlag, Berlin, 1995.