AN EXTENDABLE KNOWLEDGE-BASED SYSTEM FOR THE CONTROL OF MECHANICAL VENTILATION

Michel DOJAT,
INSERM Unité 296, Faculté de médecine, 94010 Créteil, France.
François PACHET,
LAFORIA, Institut Blaise Pascal, Université Paris VI, 75005 Paris, France.

***Abstract* - We describe a knowledge-based system that automatically controls the assistance provided to a patient suffering from respiratory failure. Our system works in a real-time closed-loop, taking physiological data from monitoring devices as inputs and acting on a ventilator connected to the patient. Our representation paradigm, based on object-orientation and production rules, is designed to support future extensions and refinements and to allow reusing of knowledge bases.**

I. INTRODUCTION

A. The medical problem

Providing mechanical assistance to a patient yields two problems: (1) adapting the assistance to the physiological needs of the patient and (2) detecting patients candidates for *weaning*, i.e., who can be disconnected from the machine. However, the clinical staff cannot continuously monitore the patient to adapt the mechanical assistance. In order to provide an assistance more adapted to the patient, and to avoid repeated disconnections / reconnections to the ventilator, we propose a closed-loop system which *continuously* adapts the assistance, and detects candidates for weaning. It is based on the expertise of the clinical staff of Henri Mondor Hospital (Créteil, France). A first version of the system has validated our approach by showing promising results [1].

B. Knowledge representation problems

Control algorithms which automatically maintain a physiological parameter, such as the respiratory frequency or inspired volume per minute, under or above a threshold set by the clinician have severe limitations [2]. They do not reflect the physician's attitude which takes into account the general context of the patient's ventilation and its evolution. In this respect, a rule-based approach is *a priori* better adapted to represent the clinical expertise on which the physician's attitude is based. However, a simple rule-based paradigm is not sufficient. The expertise involved deals with real world "objects", having specific behavior (e.g., complex devices like C02 pressure monitors, ventilators), and produces actions on these objects. Also, the expertise has a non trivial *control* aspect (in the sense of Artificial Intelligence), essentially because the reasoning of the clinician has a sequential nature, which is difficult to capture within a rule-based formalism. We propose to solve these problems in an unified and extendable fashion, based on object-orientation.

II. AN OBJECT-ORIENTED FRAMEWORK

A. Object-orientation and medical diagnosis

We start from a representation of the medical world, the Intensive Care Unit (ICU), based on *simulation* so as to adequately represent domain objects with their associated behavior. Object-orientation, and especially Smalltalk-80 [3] provides well-tried mechanisms to fullfill these needs: class abstraction, encapsulation, message-passing and inheritance. The object paradigm provides us with a first layer of representation that takes care of all the domain objects of the ICU (devices, monitors, ventilators, signal processors), as well as patients, clinicians and experts.

B. Adding an embedded rule-based system

The simulation layer does not provide any means for representing knowledge *about* the simulated world. We add a second layer, using a first-order forward-chaining rule-based mechanism embedded in Smalltalk (NéOpus) [4], that allows us to fully benefit from the mechanisms of object-orientation without any limitation, while adding a reasoning capability. This second layer, referred to as the *domain knowledge* layer, allows us to represent the knowledge of the expert, by production rules, such as diagnosis rules (e.g., hypoventilation or tachypnea) and action triggering rules (for instance rules that actually modify the settings of the ventilator). Those rules are interpreted in a forward-chaining scheme, by the NéOpus engine. They are integrated into the simulation layer in the sense that condition and action parts of the rules are expressed in terms of messages sent to the objects participating in the simulation.

C. Representing control

The control of the reasoning is a complex part of the expert's knowledge. In our system, control knowledge is far from trivial because it involves *sequencing*, which is inherently difficult to represent in a rule-based scheme [5], as well as the notion of

0-7803-0785-2/92$03.00 ©IEEE

alarms: actions that must be taken regardless of other pending actions or reasoning. An appropriate architecture is therefore needed to represent the *control knowledge* which directs the reasoning. The control is represented in NéOpus by an explicit *meta-rule architecture*. The main idea is to consider the control of a rule base as a process that requires expertise, and therefore to treat it in a similar fashion than domain knowledge: the activation of a rule base is described in terms of meta-rules that operate on control objects, and whose effect is to chose and trigger rules of the domain rule base. The control of the rule base is entirely expressed in terms of meta-rules. For instance, the alarm problem is solved by meta-rules that detect and fire any rules from the knowledge unit *alarms* as soon as they are fireable.

D. Inheritance for rule bases

The explicit representation of control in terms of meta-rules raises an other problem: since several meta-rules may be fireable at a given time, the control of the meta-base itself has to be specified. The control of meta-bases is solved by using a mechanism that introduces a relation of inheritance among meta-bases called *rule base inheritance*. In this scheme, the meta-base is seen and built, as a hierarchy of different meta-bases. An associated control strategy will then prefer those rules which are defined in the most specific (i.e., lowest) level of the hierarchy.

Each meta-base of the hierarchy deals with a particular control problem (e.g., sequencing the rules in knowledge units, firing rules in parallel, handling alarms). Alarm meta-rules are grouped in the lowest meta-base of the hierarchy so as to be preferred by the control strategy.

III. EXTENSIBILITY OF THE SYSTEM

Our approach, based on a triadic paradigm: objects for simulating the world, rules for expressing domain knowledge, and meta-rules for expressing control, is extendable without breaking the consistency of the system, for three reasons:

A. Refining domain classes with inheritance

Refining or extending the simulation layer must not break the consistency of the domain knowledge layer built on top. Within our object-oriented representation, the natural means of expressing refinements of domain concepts is class inheritance. For example, defining new devices (with specific communication protocols) is done by subclassing. Also, the good integration of the rule-based mechanism in the simulation layer allows the standard use of class inheritance to be taken into account by the rules.

For instance, it is important to be able to represent the variations on the definitions of thresholds, which values may vary, depending on the clinical practices. We will express such variations by creating subclasses that redefine only threshold methods. The rules that use those methods will not have to be changed, while remaining consistent.

B. Reusing rule bases

The rule base inheritance (an idea generated naturally by the blending of rules in to an object-oriented paradigm) introduces the notion of *rule base reuse*. A rule base may be defined as a sub-base of an existing rule base, and may be refined by a potential sub-base. This notion is particularily interesting to avoid costly rewriting of knowledge bases, and provides knowledge enterer with initial knowledge bases to be refined, rather than writing everything from scratch.

C. Reusing control and meta-bases

Last, the explicit control architecture makes it possible to also reuse control specifications (meta-bases). Control meta-rules are especially tricky to tune up. Being able to reuse a control specification is therefore of extreme help. Our achitecture provides a first step in this direction. The meta-bases we have built will be reused to control future domain rule bases that deal with similar expertises.

IV. CONCLUSION

The idea of representing medical knowledge by starting from a *simulation layer* provided by an object-oriented system, and then by adding representation paradigms for expressing knowledge (*rules and meta-rules*) yields properties of extendibility and "programming by refinements" that other systems have not. We will now use our system to represent expertises involved with automatic control of ventilators for other modes of ventilation used in ICUs and anesthesia departments.

REFERENCES

[1] M. Dojat , L. Brochard, A. Harf, "A knowledge-based system for the management of the weaning procedure of mechanically ventilated patients," *Proc. of 12th Int. Symposium on Computer Assisted Decision Support and Data Base Management in Anesthesia Intensive care and Cardiopulmonary Medecine*, Rotterdam, October 1991.

[2] F. Boyer, B. Bruno, P. Gaussagues, S. Jas-Lamonnery , D. Robert, "Aide inspiratoire avec asservissement du niveau de pression : volume ventilé minute versus fréquence respiratoire," *Réan. Soin. Intens. Med. Urg.*, vol. 5, n° 4, pp. 227-232, 1989.

[3] A. Goldberg, *Smalltalk-80: The interactive programming environment*. Addison-Wesley, 1984.

[4] F. Pachet, "Reasoning with objects: the NéOpus environment," *Proc. of EastEurOOpe conference*, Bratislava, September 1991.

[5] F. Gomez, B. Chandrasekaran, "Knowledge organization and distribution for medical diagnosis," *IEEE Trans. on systems, man, and cybernetics*, vol. 11, pp. 34-42, 1981.