SmartCare™ - Automated Clinical Guidelines in Critical Care

Stefan Mersmann[1] and **Michel Dojat**[2]

Abstract. In critical care environments important medical and economical challenges are presented by the enhancement of therapeutic quality and the reduction of therapeutic costs. For this purpose several clinical studies have demonstrated a positive impact of the adoption of so-called clinical guidelines. Clinical guidelines represent well documented best practices in health care and are fundamental aspects of evidence-based medicine. However, at the bedside, such clinical guidelines remain difficult to use by the clinical staff. Recently, we have designed and implemented the knowledge-based SmartCare™ system that allows automated control of medical devices in critical care. SmartCare™ constitutes a clinical guideline engine since it executes one or more clinical guidelines on a specific medical device. The underlying methodology comprises two sequential phases and seamlessly combines knowledge engineering with expert system techniques, e.g. rule-based forward chaining and temporal reasoning, for clinical guidelines modelling and software engineering techniques for source code generation and for integration to the target platform. SmartCare™ was initially applied for the automated control of a mechanical ventilator and is currently being evaluated in a European multicentre clinical study started two years ago. Intermediate reports have been extremely positive and suggest a statistically significant reduction in the duration of mechanical ventilation using SmartCare™. The methodology allows SmartCare™ to be implemented effectively with other medical devices and/or with other appropriate guidelines. In this paper we report on the methodology, architecture and the resulting versatility of SmartCare™ for the automated execution of clinical guidelines. Benefits and lessons learned during its development are discussed.

1 INTRODUCTION

The continuously existing and steadily increasing pressure of growing costs is being faced by an instrument known as process engineering, also within the field of medicine. The focal point of process engineering is cost saving by identification, organisation, optimisation and standardisation of business procedures. Amongst other aspects, process engineering offers many benefits related to quality assurance and quality improvement. In this way business procedures become repetitive in a controllable manner, and can be easily documented and assessed in many ways, just to mention a few benefits. For this purpose and especially in medicine so called evidence-based Clinical Practice Guidelines (in short Clinical Guidelines, CG) have been successfully established. A clinical guideline is a highly matured therapeutic plan that compiles optimum practices for treating patients in a well-defined medical context. The Institute of Medicine (www.iom.edu) defines a CG as follows: "systematically developed statements to assist practitioner and patient decisions about appropriate health care for specific clinical circumstances". CGs are therefore an effective and popular means for standardising health care practices. CGs in many ways are comparable with software patterns, which also reflect the optimum practices for solving specific requirements in software construction. In the mean time, manually executable CGs have now become available for almost all areas of health care. As an example, the National Guideline Clearinghouse™ (www.guideline.gov) at present holds 552 CGs in their archive while non-profit organisations such as the Guideline International Network (www.g-i-n.net) or their German subsidiary organisation, the Ärztliche Zentralstelle für Qualitätssicherung in der Medizin (www.leitlinien.de), are intensively building and maintaining databases for CGs.

The adoption of CGs in health care leads to a number of benefits regarding both clinical and financial outcomes:

- Optimisation of the therapy
- Improvement of therapeutic quality
- Reduction of medical device induced complications
- Reduction of hospital stay
- Reduction of mortality rate
- Alleviation of burdens on clincial staff
- Identification of potential optimisations
- Human error avoidance and reduction
- Reduction of therapeutic costs

In critical care, several randomised clinical studies have been conducted in order to quantify these benefits, especially within the field of mechanical ventilation [1, 2, 3].

However, the practical use of CGs in an extremely strenuous environment such as health care raises several problems. A strict compliance to guidelines is not guaranteed at the patient bedside and requires a specially trained team with somebody responsible for the process and adherence to guidelines [2, 3]. The difficulty of introducing CGs in real-life lies in the fact that a computerised system must be able to offer the never-ebbing advantage of being able to replace a manual guideline-driven process. Recently, an academic knowledge-based prototype has been designed, a computerised care assistant, to oversee a continuous guideline-based weaning from mechanical ventilation [4,5]. This system was aimed mainly at the automated execution of a clinically recognised CG using a knowledge-based PC-hosted system that included an automated adjustment of the targeted medical device.

SmartCare™ assumes this approach, expanding it with an appropriate and effective technology for flexibly implementing the promising task of automating the execution of CGs with a wide range of medical devices.

[1] Dräger Medical AG & Co. KGaA, Lübeck, Germany
stefan.mersmann@draeger.com
[2] INSERM-UJF UM594, Grenoble, France,
michel.dojat@ ujf-grenoble.fr

Published:
S Mersmann, M Dojat, *SmartCare - Automated Clinical Guidelines in Critical Care*, in: R Lopez de Mantara, L Saitta, eds.,
16th European Conference on Artificial Intelligence (ECAI 2004), Valencia (Spain) Aug 22nd-27th 2004:745-749, IOS-Press

The rest of this paper essentially describes our experiences with SmartCare™. In Section 2 we indicate the principles that have guided our approach. In Section 3 and 4 the architecture and the AI based techniques used are detailed. The procedure for implementing a SmartCare™ application is explained in Section 5, i.e. the implementation of a specific CG used for the weaning of mechanically ventilated critical care patients, where ventilation was provided using a Dräger Medical EvitaXL ventilator. Lessons learned during this many-year project are referred to in all sections, particularly in the discussion section.

2 METHODOLOGY

Base practices are stipulated principles, strategies and paradigms that hold for all SmartCare™ applications and are therefore not related to any specific CG.

The overall goal of developing a system to act "smartly" in terms of reliably reproducing the cognitive behaviour of health care providers, demands thoroughly considered concepts and principles. Regulatory affairs and product risk management have both taught us that the extent of application as well as a cause-effect analysis are essential aspects for success. Moreover, CGs should not be used as substitutes for clinical judgement, but should instead complement or guide it. Thus, a first but fundamental base practice of SmartCare™ applications must be: the user exclusively having complete control over the system at all times during the entire course of automatic use, and being able to override all SmartCare™ actions at any time. In a certain sense the system should act like an autopilot, being defensive and modest in nature. In addition, all SmartCare™ actions must be permanently supervised by the regular alarming management of the targeted medical device under control. This base practice implicates that an eventual hazard to the patient should in fact be excluded as a matter of course. Together with the primary requirement of providing a CG engine that is universally applicable, such an outstanding product base practice drives a second level of base practices: the technologies for choice. Assuming that any CG can be automated by SmartCare™, the targeted medical device must provide read access for all its measured values, settings, messages and contextual states and at the same time allow remote write access to all its settings. We call this functional base practice RAWS, for Read-All-Write-Some.

The entire life cycle of a designated SmartCare™ application comprises the successive phases of *Construction* and *Operation* (Figure 1). Construction covers the actual development process starting with a sound modelling of the medical expertise, and ending with the export of the knowledge base (KB) representing the original, now computerised CG. This is presently achieved based on an expert system development kit. Operation covers incorporation and automated execution of this specific CG by SmartCare™ inside the targeted medical device itself. Both phases are inherently connected via an automated source code generation, forming an efficient and comfortable change management when combined together. Such a change management enables a prompt implementation of new additional CGs on the one hand, and a cost optimised modification of already automated CGs on the other. Since SmartCare™ per definition is a typical client system, being designed for a usually non knowledge-based host system, a suitable, long-term reliable synthesis for combining conventional software engineering methods with knowledge engineering methods has to be developed. Dedicated to such a goal, we established a tool-supported consultation of CG modelling techniques during the construction phase which we call CAKE, for Computer Aided Knowledge Engineering.

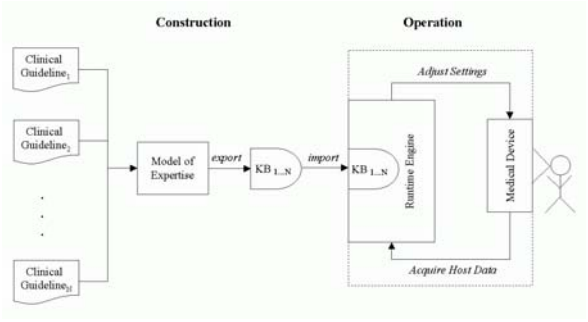

Figure 1. Automating one or more CGs: two-phase process of building and embedding into the runtime engine

The CAKE tool we decided on is the expert system development kit Solvatio® provided by IISY AG (www.iisy.com). All modelling activities of the overall system, before entering the heterogeneous knowledge engineering section of the construction phase, are documented using the unified modelling language (UML). A logical and consistent link to advanced knowledge engineering practices is supported by the use of the accepted CommonKADS methodology [6].

Comfortable and rapid implementation of other CGs must already start with the very basic modelling. Normally, the design input of a SmartCare™ application is a text-based description of the CG. Such text-based CGs often show ambiguities or are still incomplete, so that the CG author must be consulted during the initial CAKE steps. To facilitate this formalisation, we have adopted a high-level basic *Common Ontology* that serves as an invariant, skeletal template and assures the highest common ground for all automated CGs. On transition to a specific CG, all entities are specialised accordingly.

Symmetric with this static structure for the CG's model of expertise, the dynamic behaviour, i.e. the CG in execution, is also standardised to an invariant, skeletal *Common Workflow*. Consistently with the more generic therapeutic course provided by human health care providers, i.e. monitoring, diagnosis, therapy, this common workflow recurrently comprises *data acquisition*, *patient screening* and *adjustment of settings.*

3 SYSTEM ARCHITECTURE

The system architecture of SmartCare™ covers the embedded runtime system, i.e. phase operation. This has been driven by preferred attributes of quality such as flexibility, possibility for enhancement, changeability, correctness and testability. Since the SmartCare™ engine is a client application by nature, a hybrid operation of conventional and non-conventional informatics principles must be supported. This leads to the combined appliance of several architectural patterns [7] yielding to the following macro structure:

Independent Components - Self-sustaining subsystems, so called *slots*, linked to a certain transport subsystem that instantiates inter-slot communication, timer services, data streams, session handling and other aspects. For the system under discussion the following slots were defined (Figure 2):

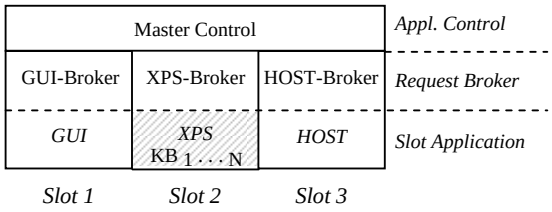

Figure 2. Macro view of the system architecture

- *GUI* – for the viewing, control and administration of the Human Interface of the application
- *XPS* – the complete expert system core comprising the knowledge base, the inference machine, dialog control and the past-oriented temporal reasoning
- *Host* – for periodic acquisition of domain relevant data from the medical device under control including pre-processing and validation of such host data

n-Tier – Each independent component, i.e. slot, is refined into three layers:

1. *Application Control* – the dynamic event-triggered control of the entire system
2. *Request Broker* – relay layer for encapsulated inter-slot access
3. *Slot Application* – the pure business functionality of the specific component in the complete system.

Virtual Machine – Finally, the *Slot Application* layer of the independent component *XPS* is designed as an interpreter/rule-base concept, allowing a well-defined and therefore inter-changeable separation between procedural and declarative functionalities.

To integrate a future component in the overall system only an additional slot needs to be opened. In the sense of object oriented methodology, each slot becomes a subsystem, which is in turn constructed as a framework of interface classes with all of the advantages of framework oriented systems. To implement a further CG for automated execution an additional knowledge base N+1 (Figure 2) had to be designed into slot *XPS*. To incorporate SmartCare™ into other target platforms entails porting or at least adopting the slots *HOST* and *GUI*.

4 AI TECHNIQUES

As a matter of definition, all AI principles of the SmartCare™ engine are encapsulated within the slot application layer of the subsystem XPS. With exactly one knowledge base inside this slot one of each CG is implemented. For the construction phase Solvatio® provides a powerful expert system shell including core AI techniques that will be inherited into the operation environment of every expert system being built with Solvatio®. These core AI techniques comprise rule-based, non-monotonic reasoning features combined with a past-oriented temporal calculus and the capability to operate multiple knowledge bases. A set of different problem solving methods like categorical, heuristic, set-covering, functional, statistical and case-comparing classification completes this construction set [8]. Combination and dynamic change between problem solving methods during runtime is possible. Defeasable, non-monotonic reasoning that allows for belief revision is implemented by an Immediate-Check Truth Maintenance System (ITMS) [9]. Temporal reasoning features are based on event-state ontology and Allen's predicate calculus [10]. A past-oriented time database using exact and relative time references as well as logic time-stamps serves as a basis for modelling temporal aspects. Finally an intuitive graphical knowledge editor enables a rapid and high-level construction of expert systems, while an integrated runtime environment offers an exhaustive case-based testing even off-line from the target platform.

The base practice for developing with Solvatio® is to select one or more problem solving methods, to model the domain ontology into a set of symptoms and a set of diagnoses, and then to integrate these using a complex network of polymorphic rules. With the recurrent propagation of this model the domain specific operation environment emerges. Theoretically, only the functional capabilities of this development kit restrict the set and scope of CGs that can be automated by SmartCare™.

The ability to handle several knowledge bases at the same time is derived directly from the primary system requirement to design a multi-CG engine. Multiple knowledge bases can be controlled in a disjoint manner or can even be shared to enable the design of CG variants that reuse a common knowledge base. Although several problem solving methods are provided by Solvatio®, only first-ordered forward chaining, i.e. categorical classification, has been used up until now in the medical applications considered. Temporal reasoning is essential for health care systems [11]. Accordingly, SmartCare™ must be able to cope with temporal entities, rules and diagnoses. For implementing the system introduced here we used the following functionality:

- Basic representation of points in time
- Relations in time relative to a given reference point
- Predefined reference points ”now”, ”start of a case” and freely definable timestamps, so-called landmarks
- Predicates for the calculation of variations over time, relations between reference points, distances in time and series of reference points

Special rules that operate on the time database together with ITMS features allow for cognitive patterns such as aggregating or forgetting, which in turn allow the realisation of typical clinical patterns such as the handling of instabilities during the course of therapy. Context sensitive timestamps reflecting ontology items are time database markers augmented with a history account. Besides enabling session tracking (*Onset of Tachypnea*), timestamps can be established by rules, compared to one another, and can be used by complex derivations or even for determining symptom values (*Duration of Weaning)*.

Figures 3 to 5 show certain graphical views into the knowledge base during the construction phase. Within the time database quantificational facts (symptoms) are tracked to retrieve certain values at a certain point of time as well as to access landmarks for reasoning purposes (Figure 3). The combined use of landmarks together with non-temporal symptoms to deduce synthesising facts is shown in Figure 4. P7 (positive) respectively N7 (negative) define a probability of 100% for establishing a rule, which indicates the categorical problem solving method. The therapeutic action after each reasoning cycle is the readjustment of the medical device under control. Hence the highest-level rules conclude all reasoning results to determine the set of new values to be set on the medical device (Figure 5). The particular name of a derived conclusion is shown in the window title.

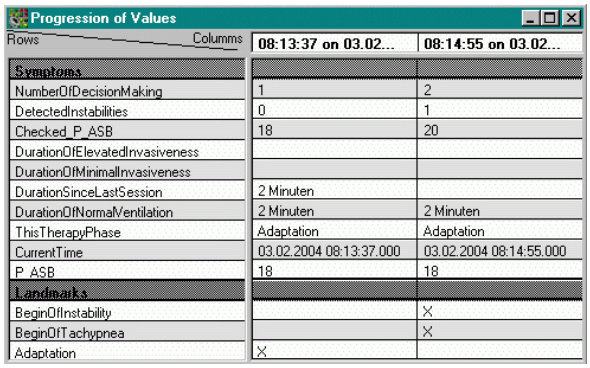

Figure 3. Construction – time database showing temporal entities and their values after two patient screenings

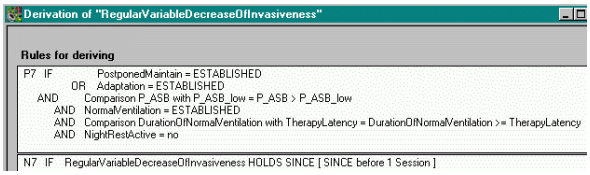

Figure 4. Construction – complex rule to derive an intermediate higher-level diagnosis

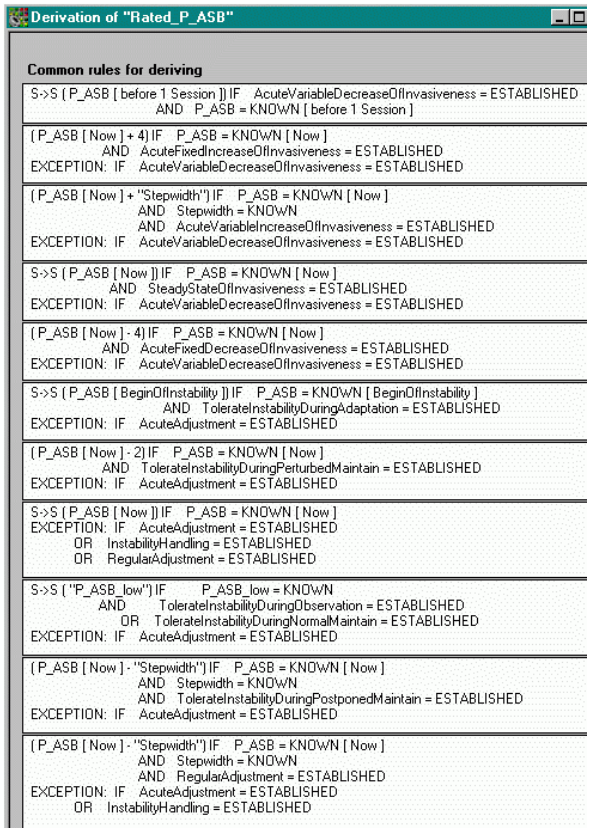

Figure 5. Construction – set of temporal reasoning rules to derive a numeric, final symptom

5 IMPLEMENTATION & EVALUATION

With the CG discussed here, SmartCare™ has been implemented for two target platforms so far: Firstly on a Windows-PC connected to an Evita 4 ventilator for evaluation and simultaneous engineering purposes, and secondly on a plug-in circuit board as the commercial product for EvitaXL. The system architecture as introduced above is implemented completely in Java: access to non-Java sections of GUI and HOST is gained via the Java Native Interface (JNI). The identification of asynchronous tasks, i.e. Java threads, is also derived directly from the slot-based architecture model. At least one thread in the system is needed for each slot to cope with concurrency. The whole functionality of the XPS slot application layer was encapsulated by a Java powered Solvatio® kernel that is identical to its CAKE version. At runtime, knowledge base, interpreter and control components are united in slot XPS.

For the CG introduced here, a total sum of 264 rules, 64 symptoms and 50 diagnoses are consulted. The layer of the *Request Broker* representing the disjointed structure of the slots was developed uniformly for all these slots. Again, Java was used.

Figures 6 to 8 are screenshots taken directly from SmartCare™ running on EvitaXL. They illustrate that SmartCare™ does not have its own predefined human interface but completely deploys the human interface of its host system, i.e. the target platform. With such a headless application, the user always operates the interface that he/she is familiar with.

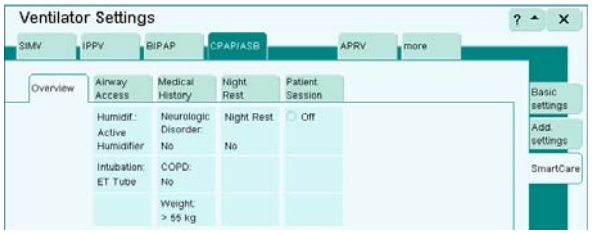

Figure 6. Operation – patient specific user inputs on start up

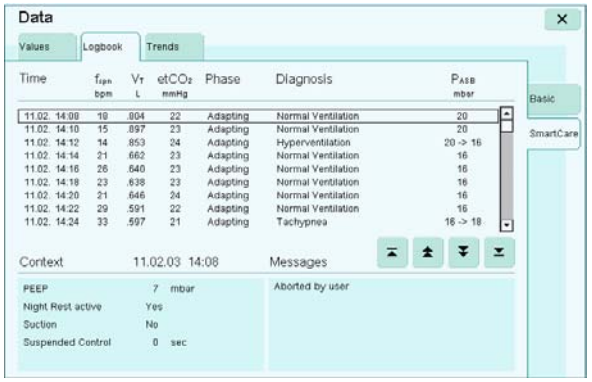

Figure 7. Operation – detailed recording of SmartCare™ patient screenings

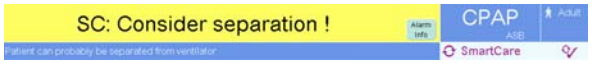

Figure 8. Operation – successful appliance of the automated CG. Separation means withdrawal from mechanical ventilation

Verification and validation of a knowledge-based system has to include procedural as well as declarative system components. All system features implemented within the slots are parts of the conventional, procedural components. The declarative parts impose a higher demand, with respect to verification and validation, than the procedural components that can be verified and validated by track-proven quality management measures such as software inspection as well as module, integration and system testing in the usual manner. Adequate clinical validation measures have already been carried out on the "NéoGanesh" system, the documentation of which proved invaluable for this undertaking [12].

As a first step, the inspection of the knowledge base together with a medical expert as guided by [13] was conducted. In order to evaluate the clinical impact of a computerised care assistant such as SmartCare™ for the management of mechanically ventilated patients, a multi-centre European study was started two years ago. The goal was to examine the effects of SmartCare™ on weaning from mechanical ventilation, especially in terms of quality of care, length of stay under ventilation, mortality and cost savings. For this purpose a total of 150 patients were randomly enrolled and ventilated using SmartCare™ or the ventilation CG in force at each centre. The complete analysis of this large mass of data has not yet been finished. Preliminary results suggest a statistically significant reduction in the duration of mechanical ventilation using SmartCare™.

6 CONCLUSIONS & DISCUSSION

There is common agreement about the wide range of therapeutic and economical benefits resulting from manually applied CGs. The automated execution of CGs by medical devices represents a logical progression of this strategy. A flexible, CG independent, multi-platform technology of an automated CG system will be able to convey this strategy. SmartCare™ provides such a technology. At present, SmartCare™ has been approved in seven European intensive care units with computerised assistants to evaluate a CG for weaning from mechanical ventilation. However, apart from the availability of appropriate technology, the sound acquisition of potential CGs is of the utmost importance. Here, a fundamental differentiation of SmartCare™ from the extensive field of knowledge acquisition must be emphasised. We experienced a huge bandwidth of CG documentation formats including simple graphical workflow diagrams or even substantial scientific text documents [14]. Fortunately, several promising approaches, especially in health care, exist for facilitating the representation, visualisation and verification of CGs [15, 16]. These guideline development frameworks are based on an appropriate representation language and frequently offer tools for efficient CG-development. XML seems to be becoming the CG representation language of first choice [17].

The results of these academic activities should be incorporated into SmartCare™ and its methodology so that the processes of CG-development and CG-execution can be coupled. Thus, CGs developed according to the above-mentioned approaches can be used as direct input for the CAKE procedure so that additional optimisations and standardisations can be implemented, entirely consistent with the idea of process engineering. During the development of SmartCare™ we have learned that the discovery of an adequate concept for linking software engineering and knowledge engineering is simplified by employing commonly used techniques such as UML and CommonKADS. Many difficulties were created and little expense was spared at the design input side so that a complete, consistent and well-documented reference CG could be obtained. The above cited approaches will be of help for this in the future. Management of restricted system resources, i.e. memory and clock-rate, is not specific to AI applications but can nevertheless become critical, especially in the division of embedded systems. Preliminary investigations and thorough system implementation resolved these issues.

REFERENCES

[1] EW Ely, AM Baker, DP Dunagan, HL Burke, AC Smith, PT Kelly, et al., *Effect on the duration of mechanical ventilation of identifying patients capable of breathing spontaneously*, N Engl J Med 1996; 335:1864–1869

[2] GP Marelich, S Murin, F Battistella, J Inciardi, T Vierra, M Roby, *Protocol Weaning of Mechanical Ventilation in Medical and Surgical Patients by Respiratory Care Practitioners and Nurses, Effect on Weaning Time and Incidence of Ventilator-Associated Pneumonia*, Chest, 2000;118:459-467

[3] SM Burns, S Earven, C Fisher, R Lewis, P Merrell, JR Schubart, JD Truwit, TP Bleck, *Implementation of an institutional program to improve clinical and financial outcomes of mechanically ventilated patients: One-year outcomes and lessons learned*, Crit Care Med 2003 Vol. 31, No. 12

[4] M Dojat, L Brochard, *Knowledge-Based Systems for Automatic Ventilatory Management*, G. Iotti, eds., Respiratory Care Clinics, Special Issue on Closed-Loop Control Mechanical Ventilation. W.B. Saunders Co., 2001

[5] M Dojat, F Pachet, Z Guessoum, D Touchard, A Harf, L Brochard, *NéoGanesh: a Working System for the Automated Control of Assisted Ventilation in ICUs*, Art Intell Med, 11,97-117, (1997)

[6] G Schreiber, H Akkermans, A Anjewierden, R de Hoog, N Shadbolt, W van de Velde, B Wielinga, *Knowledge Engineering and Management – The CommonKADS Methodology*, MIT Press, 2000

[7] L Bass, P Clements, R Kazman, *Software Architecture in Practice*, Addision-Wesley, 1998

[8] F Puppe, *Systematic Introduction to Expert Systems – Knowledge Representation and Problem Solving Methods*, Springer, 1993

[9] F Puppe, *Belief Revision in Diagnosis*, GWAI-87, K Morik (Ed), Springer (1987) 175-184

[10] J Allen, *Maintaining Knowledge about Temporal Intervals*, CACM 26, 11:832-843, (1983)

[11] Y Shahar, C Combi, *Timing is everything. Time-oriented clinical information systems*, West J Med. 1998;168:105-13

[12] M Dojat, A Harf , D Touchard, M Laforest, F Lemaire, L Brochard, *Evaluation of a knowledge-based system providing ventilatory management and decision for extubation*, Am J Respir Crit Care Med;153, 997-1004, (1996)

[13] DR Wallace, LM Ippolito, B Cuthill, *Reference Information for the Software Verification and Validation Process*, NIST Special Publication 500-234, National Institute of Standards and Technology, Gaithersburg, March 1996

[14] N Macintyre et al., *Evidence-based Guidelines for Weaning and Discontinuing Ventilatory Support*, Respir Care 2002;47(1):69-90

[15] M Peleg, S Tu, J Bury, P Ciccarese, J Fox, R Greenes, R Hall, P Johnson, N Jones, A Kumar, S Miksch, S Quaglini, A Seyfang, E Shortliffe, M Stefanelli, *Comparing Computer-Interpretable Guideline Models: A Case-Study Approach*, J Am Med Inform Assoc, 52-68, 10(1), Jan-Feb, 2003

[16] G Duftschmid, S Miksch, W Gall, *Verification of Temporal Scheduling Constraints in Clinical Practice Guidelines*, Art Intell Med, 93–121, 25(2), 2002

[17] T Dart, Y Xu, G Chattelier, P Degoulet, *Computerization of Guidelines: Towards a "Guideline Markup Language"*, MEDINFO 2001, V Patel et al. (Eds.), IOS Press, 2000