

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,500

Open access books available

118,000

International authors and editors

130M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



# Clinical Decision Support with Guidelines and Bayesian Networks

Oliver Nee and Andreas Hein  
*OFFIS – Institute for Information Technology  
Germany*

## 1. Introduction

The field of medicine has been one of the favourite application areas for tools of artificial intelligence (AI) research for decades. One of the reasons for this focus may be the fact that decision making in medicine is hard and by all means a worthy challenge. This is not only due to the countless symptoms, tests, diseases and treatment options that make up the medical body of knowledge, but also because of the complexity of the human physiology and the resulting lack of absolutes in medicine. Traditional rule-based approaches try to employ first-order logic in order to model causal or diagnostic relationships between concepts do not work for complex medical reasoning because the connection between symptoms and conditions is just not a logical consequence in either direction of a rule. Given the impracticality of logically exhaustive rules and a certain measure of incomplete knowledge - be it theoretical due to incomplete science or practical due to a lack of data, an agent striving to make a decision can at best have a degree of belief in the truth of a statement. Bayesian networks can be used to model causal (or diagnostic) relationships using that degree of belief, allowing reasoning under uncertainty.

The difficulty of clinical decision making doesn't just affect computers. The diagnosis and treatment of complex diseases such as cardiovascular diseases require a series of decisions that are often based on incomplete data. Clinical Practice Guidelines (CPGs) help clinicians make these decisions by providing recommendations that are based on various levels of evidence. However, studies have shown that guideline compliance tends to be low unless the recommendations for a specific case are made when a decision needs to be made. In the clinical practice, this would be realized as a clinical decision support system (CDSS) that is integrated with other clinical information systems and that offers case-specific advice. Currently, all frameworks for computer-interpretable guidelines face the problem of having to deal with reasoning under uncertainty due to incomplete data or incomplete knowledge. Introducing a degree of belief in data items would increase the robustness of a CDSS.

This chapter discusses a CDSS built on Bayesian networks that is part of a system designed for cardiac tele-rehabilitation for patients after a myocardial infarction. The system replaces a rule-based variant that had been used to control the patients' exercise sessions, including the generation of alarms. The evaluation shows that the number of false alarms can be reduced by the introduction of Bayesian networks.

Source: Decision Support Systems, Advances in, Book edited by: Ger Devlin,  
ISBN 978-953-307-069-8, pp. 342, March 2010, INTECH, Croatia, downloaded from SCIYO.COM

## 2. Decision support in medicine

### 2.1 Clinical practice guidelines

The Institute of Medicine defined clinical practice guidelines (CPGs) as "systematically developed statements to assist practitioner and patient decisions about appropriate health care for specific clinical circumstances" (Field & Lohr, 1992). For the sake of simplicity, clinical practice guidelines will simply be referred to as guidelines. Guidelines are created by scientifically working medical societies and exist to help doctors provide their patients with the best possible treatment. In Europe, for example, the European Society of Cardiology (ESC) provides over 30 guidelines covering various cardiological conditions and their treatment options. Guidelines are an application of evidence-based medicine (Elstein, 2004). Evidence-based medicine (EBM) makes use of tools such as meta-analysis of medical literature, risk-benefit analysis, and randomized controlled trials (RCTs) in order to qualify the level of evidence and subsequent recommendations. It should represent the current state of art in clinical medicine and aim to help the Doctors to provide the best medical care.

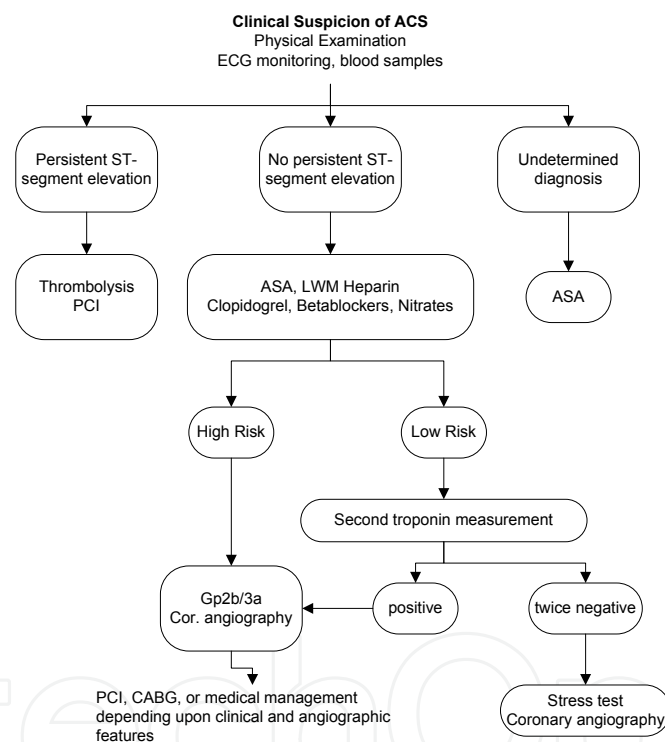


Fig. 1. Excerpt from a guideline

Fig. 1 shows an excerpt from a guideline (Bertand et al., 2002) that suggests a strategy for patients with a suspected acute coronary syndrome (ACS), a set of signs and symptoms related to the heart. Based on the results of the patient's electrocardiogram (ECG), the strategy discerns between various subtypes of the ACS and suggests suitable treatments. The simple flowchart representation of the strategy should not fool a reader into believing that a guideline is little more than that. The guideline that describes the strategy for the center branch (for patients who present without a persistent elevation of the ST-segment in their ECG) is 32 pages long and substantiates its findings with over 200 references.

Guidelines are not meant as ex cathedra statements telling Doctors how to treat their patients. Rather than that they should be understood as means of distributing knowledge, of

making good clinical practice known to those who practice it. Guidelines are being endorsed because of their contents and the way these contents are being produced by expert groups. Simply gathering and condensing knowledge into guideline documents will not benefit the patients, putting that knowledge into action does. And in order to put the knowledge from guidelines into action, they need to be aware of it. Guidelines are published in medical journals and other media. However, this knowledge is rarely available to the physician at the point of care where decisions are made. It is not realistic to expect a cardiologist to remember the recommendations of more than 30 guidelines, and referring to them for each and every case would be too time-consuming. One way to aid this decision making process is the introduction of clinical decision support systems that should ideally be integrated into the IT systems that are being used in the daily routine.

## 2.2 Clinical decision support systems

Wyatt and Spiegelhalter (1991) define a clinical decision system (CDSS) as “an active knowledge systems which use two or more items of patient data to generate case-specific advice.” Clinical Decision Support Systems help a physician by suggesting a diagnosis based on findings and/or suggesting a treatment. Diagnostic support can be especially useful for rare and complex diseases, though planning support for therapies is an even more interesting application of CDSSs, since it supports evidence-based medicine (EBM) and best practices. Programs that support diagnostic decision making have been shown to improve a physician's diagnostic performance significantly, as shown by Berner et al. (1999). Programs can also support a clinician's therapy planning, improving the quality of care, as shown in a recent trial (Field et al., 2009).

The following subsections discuss a small selection of current decision support systems. DxPlain is a decision support system that helps physicians in the diagnostic decision making process. CARDSS has been implemented to improve physicians' adherence to cardiac rehabilitation guidelines and the screening of patients. TREAT is a system designed to improve the prescription of antibiotics.

DXplain has been developed at the Laboratory of Computer Science at the Massachusetts General Hospital. Development of DXplain started in 1984 with information on 500 diseases being available in the initial release in 1986. The article (Barnett et al., 1987) is the earliest paper describing the system. DXplain can be accessed through the web, and since 1996 it is the only method of distribution, replacing the stand-alone program that was in use until then. It operates in two modes: it can serve as an electronic medical textbook, and it is a medical reference system. The knowledge base of DXplain contains information on 2200 diseases and 4900 clinical findings. Clinical findings can be symptoms, signs, epidemiologic data and laboratory, endoscopic and radiological findings. Based on the findings observed by the physician, the system will determine a ranked list of diseases as well as a detailed description of the disease. The users and developers have been fine-tuning the knowledge base for more than 20 years, so it can be assumed that DXplain offers good sensitivity and specificity. DXplain does not offer decision support that goes beyond diagnostic decision making. However, simply by suggesting additional diagnoses and findings, it can help physicians detect rare diseases they would usually not have considered.

CARDSS stands for "Cardiac Rehabilitation Decision Support System" and is a computer-based guideline implementation system for cardiac rehabilitation screening (Goud et al., 2005). It was developed concurrently with recently released guidelines of the Netherlands

Heart Foundation. The decision support system has been developed in order to improve the assessment of a patient's need for cardiac rehabilitation. A major part, and in fact the main focus of CARDSS, is the screening procedure that can be described as flowchart containing 15 to 30 questions to be answered, covering the patient's clinical history as well as the current situation. The purpose of the screening procedure is to determine whether or not the patient should be considered for cardiac rehabilitation, and if so, what the patient's goal of the rehabilitation should be and what therapy should be chosen. A large factor that contributed to the success of CARDSS during its pilot study is the concurrent development of the new cardiac rehabilitation guidelines and CARDSS so that the paper-based guidelines and the decision making in the DSS mimic each other.

TREAT (Leibovici et al., 2007) is a decision support system for antibiotic treatment in inpatients with common bacterial infections and is based on a causal probabilistic network and uses a cost-benefit model for antibiotic treatment, including costs assigned to future resistance. The system has been tested in a cluster randomized trial (Paul et al. 2006). According to the study, the length of hospital stay, costs related to future resistance and total antibiotic costs were lower in intervention versus control wards, which leads to the conclusion that TREAT improves the rate of appropriate empirical antibiotic treatment while reducing antibiotic costs and the use of broad-spectrum antibiotic treatment.

### **2.3 Computer-interpretable practice guidelines**

The systems discussed in the previous section are specifically tailored systems that are isolated from the information systems usually used in clinical practice. The process of transforming a paper-based guideline into a computer-interpretable format and adapting it to local workflows is time-consuming; that is why approaches that allow the sharing of computer-interpretable guidelines are being sought. For this purpose, various frameworks for computer-interpretable guidelines (CIGs) have been developed. However, Sonnenberg and Hagerty (2006) reported that "at this time, there is no dominant CIG framework and no system that is in widespread clinical use outside the institution in which it was developed", and this assessment still holds to date. Comparing all recently proposed and implemented frameworks would be beyond the scope of this section. Other articles describing and comparing current frameworks are the following: Peleg et al. (2003) compared six guideline models: Asbru, EON, GLIF, GUIDE, PRODIGY, and PROforma. Sonnenberg and Hagerty (2006) discuss the state of art and also compare Arden Syntax, GLIF, Asbru, PROforma, EON, GUIDE, PRODIGY and DILEMMA/PRESTIGE. Isern and Moreno (2008) compared Arezzo, DeGeL, GLARE, GLEE, HeCaSe2, NewGuide, SAGE and SpEM. Another comparison of five frameworks can be found in de Clercq et al. (2004), where the Arden Syntax, GLIF, PROforma, Asbru, and EON are discussed. This section will focus on a format called GLIF, the Guideline Interchange Format, since it is a typical framework.

The Guideline Interchange Format (GLIF) was developed by the InterMed Collaboratory and is a collaboration of Stanford Medical Informatics, Harvard University, McGill University and Columbia University. The first published version of GLIF was released in 1998 (Ohno-Machado et al., 1998). As the name suggests, the aspect of sharing guidelines between various medical institutions was one of the design goals of GLIF. This intended sharing is supposed to work for humans (who can read the guidelines) and for automatic parsing by computers and integration in a clinical decision support system. Guidelines in GLIF should be (1) human-readable, (2) computer-interpretable and (3) share-able and thus adaptable by different institutions. These aims are reflected by the three-layered approach of

GLIF. This layered approach of GLIF also helps facilitate the collaboration process required for the transformation of paper-based guidelines into a computer-readable format that is integrated into the clinic's electronic patient record. The conceptual layer is the layer experts from the medical domain are mostly involved with. On the conceptual level, a GLIF guideline consists of a sequence of steps that are linked in a directed graph that can be represented as flowchart. The flowchart representation of a small portion of a guideline is shown in Fig. 2.

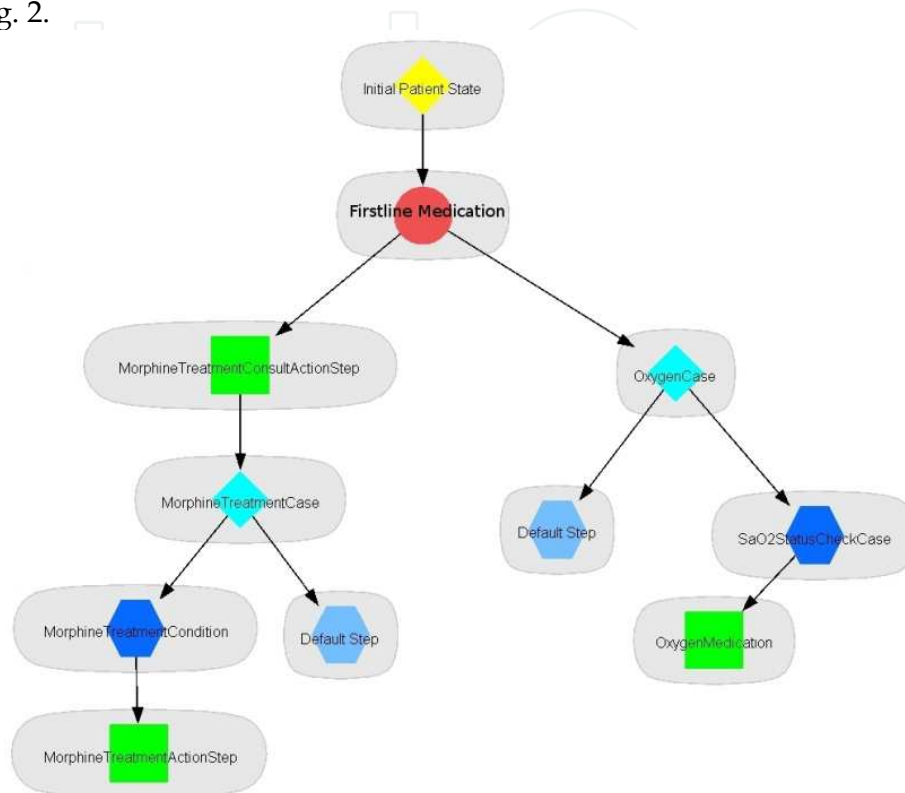


Fig. 2. Flowchart representation of a GLIF-encoded guideline

However, this sequence is not executable yet. The computational layer of GLIF is used to provide the necessary definitions and formalizations. The third layer, the implementation layer deals with the integration of the formalized guideline into the flow of clinical data. Here concepts from the guideline are mapped to clinical data that is retrieved from various clinical information systems. The elements in this layer may be non-sharable.

GLIF defines five steps: decision steps, patient state steps, branch steps, synchronization steps, and action steps. As the name suggests, decision steps are used at points in a guideline if there is more than one possible step to follow the currently active one. In Fig. 2, the step labeled 'OxygenCase' is such a step, where a decision is made whether the patient should receive oxygen or not. There are two kinds of decision steps, case steps and choice steps. 'OxygenCase' is a case step, which means that it contains at least one logical expression that determines the further flow of the guideline through the various alternatives following the case step. In the example, the further steps depend on the evaluation of the logical expression that compares the patient's oxygen saturation against a specified threshold. Choice steps are used to represent a situation where the further execution of the guideline depends on a choice being made by an external agent (for example a clinician using a user interface) based on preferences suggested by the guideline. Choice steps contain rules that support or oppose the various preferences.

Patient state steps can be seen as labels that are used to describe the current patient state as it is expected by the guideline after having executed the previous steps. A patient state step can also be used as an entry point in the guideline, depending on the current patient's state. The patient's state is described with attributes that are contained in the patient state step. If used as an entry point, the guideline is executed when criteria match the attributes.

Guidelines may require the concurrent execution of clinical tasks (i.e. running lab tests while waiting for the results of an X-Ray). Branch steps are used to model this concurrency and are used in conjunction with synchronization steps. Once the corresponding synchronization step is reached after a number of steps have been performed within a branch, a logical expression serving as continuation attribute specifies whether all, some, or one of the preceding steps must have been completed before control can move to the next step. Actions steps are used to represent medical tasks that have to (or should) be performed.

### 3. Decision support in a homecare scenario

Consider a treatment scenario that is aimed toward cardiac patients after successful treatment and rehabilitation. It takes place at a patient's home – which is why we shall refer to it as 'homecare scenario' – and is designed to facilitate a safe, medically supervised training. The rationale behind this scenario is a reduction of the length of the costly in-patient phase of the rehabilitation, and more importantly, the reduction of re-hospitalizations through better secondary prevention. A scenario like this is realized with a software-hardware system. Aside from device that captures the patient's electrocardiogram, the patient is also equipped with a pulse oximeter that measures the oxygen concentration in the patient's blood and a blood pressure sensor. As a training device, the patients use a modified ergometer bike that has been fitted with a Panel PC. This ergometer bike and the sensors can be seen in Fig. 3-a. The Panel PC with its touch screen not only serves as user interface for the patient but also as a gateway that gathers the data from the sensors, controls the ergometer load and connects the patient's home with the rehabilitation clinic so training plans can be updated and training sessions be monitored remotely.

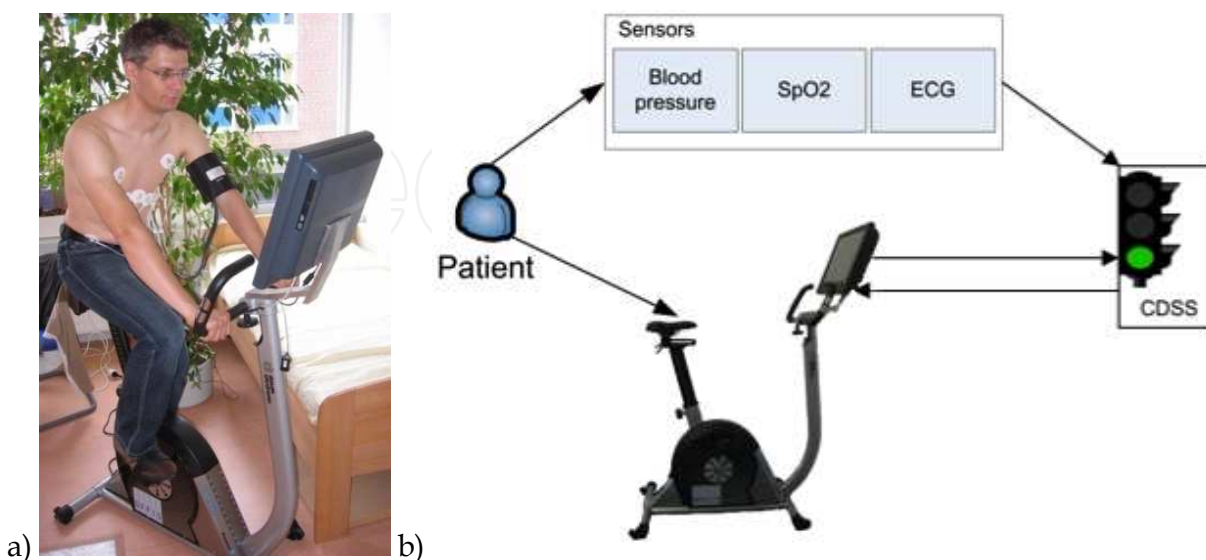


Fig. 3. a) Modified ergometer bike and sensors used in the homecare scenario and b) Flow of sensor and actor data to CDSS

However, in order to ensure the patient's safety during the training sessions, the software needs to go beyond the simple streaming of sensor data to the rehabilitation clinic. The data needs to be analyzed locally, through a sensor flow shown in Fig. 3-b, allowing the system to adapt the training load or abort the training session if the patient's state is no longer stable.

The patient's state - and with it the status of the running training session, can be represented with a traffic light schema that is easily understood by patients and doctors alike. A green light means that the session can commence or continue because the patient's state is stable. Since the training program is tailored to the patient's treatment and rehabilitation progress, this is the expected state. The yellow light state is used if the patient's vital parameters exceed the expected thresholds but do not pose a danger to the patient. In that case, the training can continue, though the training load may be lowered to keep the patient in a stable state until the end of the training session. However, a red light signals the end of the training session because the patient's vital parameters exceed the set thresholds to such a degree that the patient may be in danger if the training session is continued. Adverse events like that are rare but need to be treated with care. The patient could be in danger and should see a doctor as soon as possible, and the treating physician should be notified of the alert and seek contact with the patient. The patient can also signal problems by pressing a corresponding red or yellow button on the graphical user interface. The mock-up of a typical interface in a homecare scenario is shown in Fig. 4. The interface was developed for the OSAMI-D project by C-LAB, a research institute of the University of Paderborn and Siemens SIS.

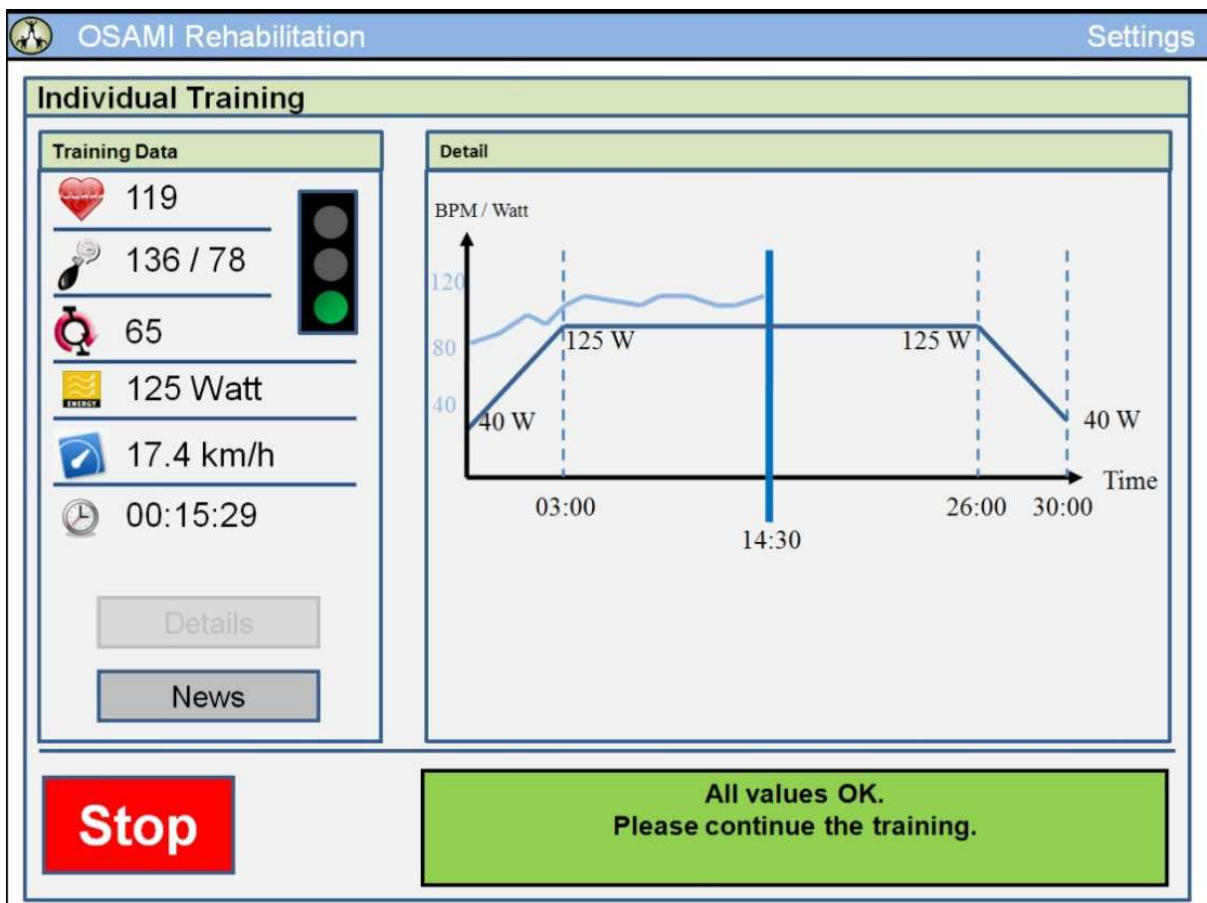


Fig. 4. User interface for a homecare application



The simplicity of the traffic light schema hides the fact that analyzing the sensor data is not a trivial task; dealing with simple data items such as the blood pressure, the blood oxygen saturation or the heart rate poses less of a challenge than the signal processing and analysis of ECG signals. The continuous analysis of ECG signals, however, is a vital step in ensuring the patient's safety.

A concept for decision support in a homecare scenario is shown in Fig. 5. The homecare system is a distributed approach between the clinic's side and the system deployed in the patient's home. In the clinic, an individual training program is created for the patient by adapting the generic exercise guideline for the patient, taking into account data gathered during the therapy, the inpatient rehabilitation, a number of reference training sessions and the results/reports of the last sessions performed at the patient's home. This adapted guideline is downloaded by the system in the patient's home at the start of the training session and is used to determine the parameters of the patient stabilization loop and for the generation of alerts. One possible implementation of the components realizing the stabilization loop and alert generation is described in the next section.

#### 4. A rule-based approach

Nee et al. (2008) describe a system that realized a system like the one shown in Fig. 5. At the patient's home a CDSS is implemented that uses a rule-based approach for patient stabilization and alert generation. Before this actual implementation is discussed, the patient stabilization loop needs to be detailed further.

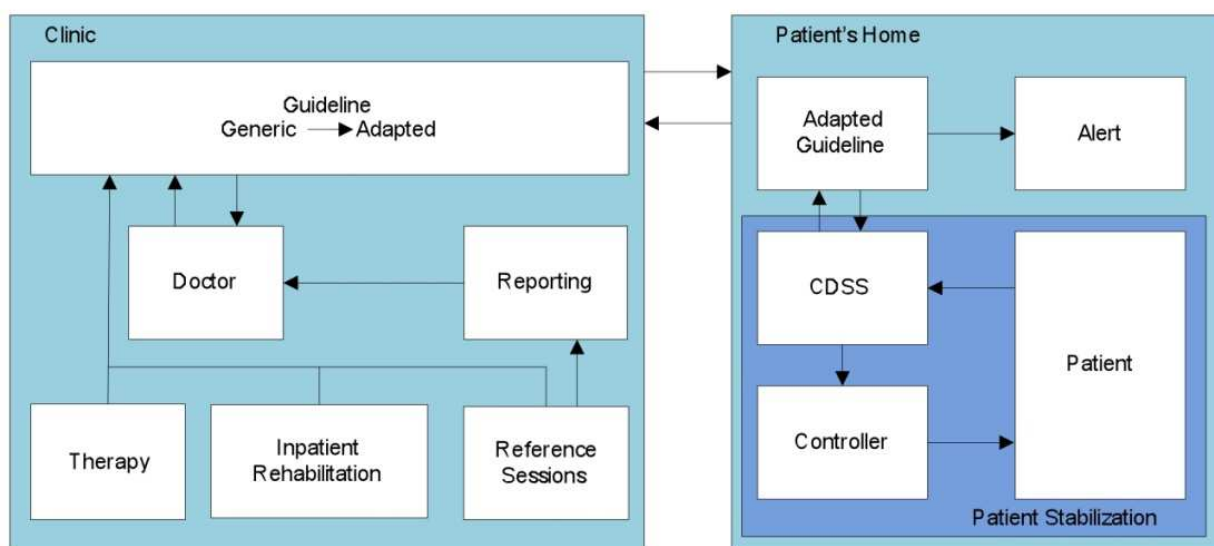


Fig. 5. Concept for decision support in a homecare scenario

##### 4.1 Control loop for patient stabilization

The control loop for the patient stabilization during the training at home is shown in Fig. 6. It consists of the patient, the ergometer, a CDSS, and a controller. The patient is observed by the sampled sensor values  $y_p(k)$ , that consist of blood pressure (BP), heart rate (HR), oxygen saturation (SpO<sub>2</sub>), and the ECG signal acquired via 4 or 10 defined positions on the patient's body (3/6/12 channel-ECG). Depending on sensed values  $y_p(k)$ , the power  $P_p(k)$  applied by the patient to the ergometer (measured and transmitted as  $y_e(k)$ ) and initial parameters and

thresholds the CDSS derives the patient state  $x(k)$  and the control difference  $e(k)$ . The control difference  $e(k)$  is used continuously by the controller to derive control output (increase or decrease of brake force) to the ergometer  $u_e(k)$  and to the patient via visual display  $u_p(k)$ . In this way the training can be defined either as power/load over time (see Fig. 4) or HR over time. In addition, the discrete patient state  $x(k)$  derived from the CDSS accordingly to predefined rules is used to switch between different states of a training session, to initiate alerts and to report the training results to the clinic.

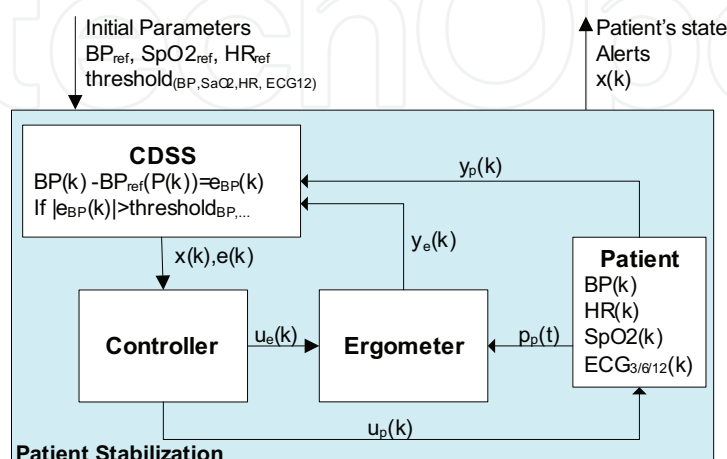


Fig. 6. Control loop for patient stabilization

The patient stabilization is defined to reach the following aims:

1. *Improvement of patient's compliance to the predefined training plan:* Because of safety reasons there is only a weak coupling between ergometer and patient. That means, that neither the training itself nor the power transmitted from the patient to the ergometer during the training can be controlled directly. Due to this, the parameter HR (an in principle also BP) can be used as input to the control loop and can be stabilized automatically by varying the ergometer's brake power.
2. *Determination of potentially critical states:* According to [Gi02] absolute, the differential and the relative thresholds can be used to derive the patient's state  $x(k)$  from the observed parameters  $y_p(k)$  to control the training:
  - Absolute thresholds are defined independent on the training state of the patient. Example is the oxygen saturation that should not fall below a certain level, e.g.  $(SpO2) < 90\%$ . These levels can differ from patient to patient and have to be defined based on the initial training in the clinic.
  - Differential and relative thresholds depend on one hand on the training state  $y_e(k)$  of the patient (e.g. blood pressure (BP) before the training start should  $< 200/110$  mmHg and during the training  $< 250/115$  mmHg). On the other hand either the change of one parameter should not exceed a threshold (e.g. drop of BP by 10 mmHg despite increased workload) or the chronological sequence of one parameter should not be abnormal in respect to the reference curves recorded during the initial training in the clinic (e.g. respiration).

Depending on the identified deviation  $e(k)$  and classified patient's state either the training is eased by opening the ergometer's brakes and informing the patient to slow down or the training will be interrupted and the clinic will be informed via alert channels.

3. *Identification of technical problems:* The detection of sensor failure or erroneous sensor placement is of special importance to avoid false alerts. Critical are problems during the placement of the sensors (especially ECG electrodes) and during recording and interpretation of ECG and blood pressure because these parameters are used as indicators for the termination of the training.

#### 4.2 Implementation of the rule-based approach

The model discussed in section 4.1 was realized using the rule engine Jess and a number of rules that examine the vital parameters, generating alert events if they exceed thresholds. Jess has been chosen because of its execution speed and the easy integration in Java. A Jess rule has a left hand side (LHS) that determines when a rule is activated and a right hand side (RHS) that describes what actions will be executed if the conditions defined in the LHS are met. Consider the following rule for the detection of high systolic blood pressure:

$$\begin{aligned} & (\text{bpmeasurement } \{\text{systolicP} > 200\}) \\ & (\text{not } (\text{alarm } \{\text{sensor} == \text{BP} \ \&\& \ \text{severity} > 1\})) \end{aligned} \quad (1)$$

$$\begin{aligned} & (\text{assert } (\text{alarm } (\text{patientID } ?\text{patID}) (\text{sensor } \text{BP}) (\text{alarmType } \text{bptoohigh}) (\text{date } ?\text{date}) \\ & \quad (\text{time } ?\text{time}) (\text{severity } 2) (\text{alarmMessage } \text{"BP} > 200\text{mmHg"}))) \\ & \quad (\text{alarm-to-db } \text{"BP} > 200 \text{ mmHG"} \text{ BP } 2) \end{aligned} \quad (2)$$

The conditions of the LHS are contained in (1), the RHS in (2). Every alarm has a severity, ranging from 0 ("green") to 2 ("red"). So the LHS says that the rule should be triggered if the systolic blood pressure is higher than 200 mmHg and if there hasn't been a previous red alert caused by high blood pressure. The RHS generates the actual alert messages, one that is sent to other components, such as the training controller that stops the training at a red alert, and one that is written to the database for reporting.

A simple rule like the one shown in (1) and (2) is too likely to be triggered by artefacts in signals that are susceptible. For data items that are generated every second (such as data coming from the pulse oximeter), a counter helps mitigate this problem. For the heart rate and oxygen saturation measured by the pulse oximeter, it takes three consecutive measurements to trigger an alert.

Since Doctors generally do not want to write rules using a rather complex syntax, the customization of the rules is performed through a graphical user interface that allows thresholds to be changed at any point during the training. In order to reflect the dynamic nature of a training session, at least three sets of thresholds need to be defined: the first set decides if the actual training can commence, and the other two sets are used during the warm-up phase and the cool-down phase.

#### 4.3 Evaluation of the rule-based approach

While the rule-based approach appears to be the easiest and most natural one, it has displayed some shortcomings when the alarm generation system was tested with patients. The first shortcoming has already been mentioned in the last section: even simple comparisons require complex rules with functions that are not obvious, which makes it hard for medical doctors to validate the behavior of the system. Debugging these rules is not trivial. These weaknesses could be mitigated by using a simplified domain-specific language (DSL) that exposes only a subset of the Jess language.

However, the performance of the alert system was not satisfactory. Four patients - three men, one woman - ( $38.5 \pm 18.5$  years,  $183.5 \pm 7.8$  cm,  $77.5 \pm 7.8$  kg, BMI  $23.5 \pm 1.3$  kg/m<sup>2</sup>) agreed to participate and test the bicycle ergometer. The patients trained on  $9.75 \pm 1.71$  days, and 42 training reports were generated. A training report was only generated if sensor data has been recorded. There may have been attempts to start training sessions that were aborted because of issues with the sensors or for other reasons. If difficulties occurred, they have been recorded. No serious events were caused by the heart disease, but a number of minor difficulties regarding sensor operability.

Despite the number of events recorded for each patient, the system worked as expected. However, the results of the patient are a good starting point for a discussion on the sensor data that needs to be dealt with in a realistic setting. Fig. 7 shows a 10 second segment of ECG data that is typical for ECG data with motion artefacts. Reliable automatic interpretation of such signals is very hard. The signal quality and the capabilities of the ECG analysis software that was used in the project allow for rhythm analysis at best, and quite often not even that.

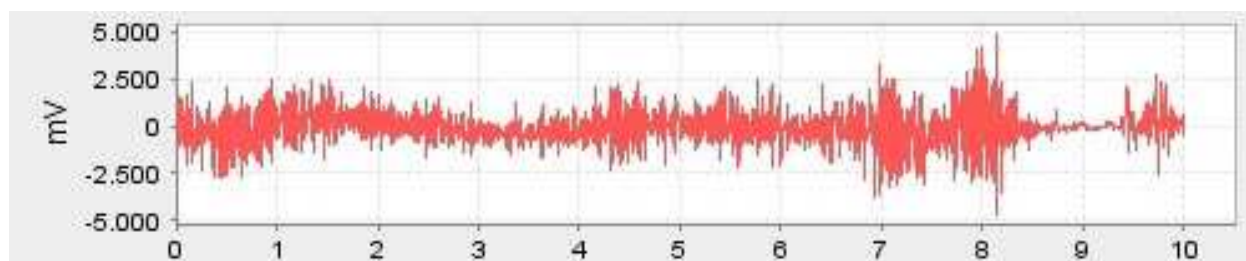


Fig. 7. ECG signal with motion artifacts

The vast majority ( $\geq 95\%$ ) of events documented in the reports are ECG-related. At this point it is important to remember that the physicians who supervised the training sessions did not find any major events, and the patients did not report any physical discomfort. Also, the training sessions were completed within two weeks, so no drastic changes in the ECG results are to be expected. This is why the high number of ECG-related alerts is most likely due to misinterpretations from the analysis software.

So while this approach allows for an easy specification of alarm conditions and can be mapped to the traffic light schema used in the homecare scenario, it is an agnostic one: the analysis of the vital parameters and signals at a given discrete time point  $k$  does not build on the results from  $k-1$ . For data gathered by the pulse oximeter, alarms are only triggered if the specified threshold is exceeded by three consecutive thresholds. While this method prevents a great number of false alarms, it treats each measurement result with the same degree of belief and thereby in fact amplifies the effect of motion artifacts on the generation of alarms. The strict Boolean approach of rules does not consider concepts like uncertainty and belief and cannot be used to capture the probabilistic nature of medical domain knowledge.

## 5. Clinical practice guidelines and Bayesian networks

One formalism that has been shown to be suitable for dealing with reasoning under certainty are probabilistic networks called Bayesian Networks (BN), which are also known as Bayesian Belief Networks or Bayes Nets. In the remainder of this chapter, either the term Bayesian Networks or its acronym will be used. Covering Bayesian networks and their construction in depth would be well beyond the scope of this work. A more thorough

treatment can be found in the seminal work of Pearl (1988) or the more accessible Neapolitan (2003).

A Bayesian network  $B = (Pr, G)$  is a joint (or multivariate) probability distribution over a set of random variables with a graphical structure  $G$  and an associated probability distribution  $Pr$ . The graphical structure is a directed graph  $G = (V(G), A(G))$  with nodes  $V(G) = \{V_1, \dots, V_n\}$ ,  $n \geq 1$  and arcs  $A(G) \subseteq V(G) \times V(G)$ . Each node  $V_i$  represents a random variable having a finite set of mutually exclusive states. The graph is acyclic, which means that there is no directed path  $V_1 \rightarrow V_n$  so that  $V_1 = V_n$ . The arcs are used to model the probabilistic influences between the variables. The parents of a variable  $V_i$  are denoted as  $\Pi(V_i)$ . The joint probability distribution  $Pr$  is given in a factorised form. For each variable  $V_i$ , there is a specified set of probability distributions  $Pr(V_i | \Pi(V_i))$  describing the joint effect of a specific combination of values for the parents  $\Pi(V_i)$  of  $V_i$  on the probability distribution of  $V_i$ . The joint probability distribution of  $V_1, \dots, V_n$  is given in (3).

$$Pr(V_1, \dots, V_n) = \prod_{i=1}^n Pr(V_i | \Pi(V_i)) \quad (3)$$

Due to the Markov condition, the number of probabilities necessary for the joint probability distribution grows in a linear fashion  $O(n)$  rather than exponentially  $O(2^n)$ , resulting in less effort for specification and computation. As long as they are acyclic, the graphs representing Bayesian networks can be arbitrarily complex and make use of nesting. Classification problems can usually be solved with a specific class of networks that have a limited topology.

Bayesian networks can be constructed manually or learned from data. Through learning, the topology of the Bayesian network and the joint probability distribution can be constructed, or the learning process can be used to generate the underlying probability distribution. Since the joint probability distribution grows in a linear fashion rather than an exponential one, allowing even complex causal interaction between many variables to be specified and interpreted with reasonable effort. The graphical representation makes it easy for domain experts to create and interpret a Bayesian network for a part of their domain.

Aside from its topology, a Bayesian network requires the specification of the joint probability distribution. This is usually done via a conditional probability table (CPT), which is defined for each variable, taking into account the probabilities of the parent variables. The CPTs can be specified by domain experts, or they can be learned from data. Learning the topology and the CPTs from data is also possible but is computationally expensive because of a huge search space. The flexibility of how a Bayesian network can be constructed and refined is one of the distinct advantages of this approach.

### 5.1 Bayesian networks in medical applications

Medicine has so far been the most popular field for the application of Bayesian networks. Given the complexity of the domain, the amount of knowledge (often implicitly) held by medical practitioners, there has long been a strong interest in expert systems that aid medical personnel. The ability to explicitly model causal dependencies and their simple and effective visualization through nodes and vertices are part of the appeal of Bayesian Networks. Lucas et al. (2004) identified four important types of problem solving where Bayesian network methods have been applied: diagnostic reasoning, prognostic reasoning and treatment selection, and the discovery of functional interactions.

When constructing a diagnostic hypothesis and during the elimination process, the inherent uncertainty of diagnostic testing needs to be taken into consideration in order to prevent a misdiagnosis. This type of reasoning with uncertainty can easily be modelled with Bayesian networks. PATHFINDER (Heckerman et al., 1992) is an expert system for the diagnosis of lymph node diseases. Another early example is the MUNIN network (Andreassen et al., 1987) for the interpretation of electromyographic findings. A more recent application is the BayPAD network (Luciani et al., 2003). The acronym 'BayPAD' stands for 'Bayesian network for Pulmonary embolism Assisted Diagnosis'. The network for the diagnosis of pulmonary embolism includes 72 random variables that represent both the risk factors and the pathophysiological consequences of the disease, which can be detected by diagnostic tests. Diagnostic reasoning performed with Bayesian networks can also be exploited for e-learning purposes, as presented with CardioBayes (Seebold et al., 2005). CardioBayes was designed as a learning environment for students and is based on a Bayesian network design by domain experts. One example for prognostic reasoning is NasoNet (Galan et al., 2002), which extends a Bayesian network model with temporal reasoning. TREAT, a system for treatment selection, has already been introduced in section 2.2. The selection of treatments requires a combination of diagnostic reasoning and prognostic reasoning. Decision support systems often use Bayesian networks as a basis when suggesting possible treatments. However, it is also possible to extend the Bayesian network model in order to capture concepts like preferences and costs that are necessary for decision support. Influence diagrams (Owens et al., 1997) can be used for this task.

## 5.2 Application in the homecare scenario

To address the shortcomings of a guideline modelling language like GLIF and in order to add robustness through reasoning under uncertainty to guidelines during their runtime, a combination of Bayesian network techniques and GLIF is being sought.

The core of a guideline-based training control implements an alarm system for a Homecare Scenario based on Bayesian networks, building on the results of the well known ALARM Bayesian network (Beinlich et al., 1989). The resulting system will be compared with the rule-based alarm system described in section 4.2, which uses a set of rules written for and executed by the Jess engine. Using thresholds defined by the clinician for each patient, the conditional probability tables (CPTs) of the Bayesian net will be refined and tested with data from actual training sessions.

Fig. 8 shows the basic outline of a guideline that can be used to control a training session in a homecare scenario. The decision step "Training OK?" and the action step "Training" use Bayesian networks to determine if the patient's state allows the training session to commence or to continue.

## 5.3 Implementation

The first step of the implementation is the extension of the Guideline Interchange Format to incorporate Bayesian networks. Fig. 9 shows the extended format, with the actual extensions being limited to the classes BayesNet\_Node, BayesNet\_Criterion and Meta\_Patient\_State.

The class BayesNet\_Node captures the topology and probabilities of a Bayesian network, while the class BayesNet\_Criterion makes it available for the decision making process in GLIF decision steps. The class Meta\_Patient\_State is used to store a Bayesian network that might be queried by more than one decision step. Although it would be possible to use these

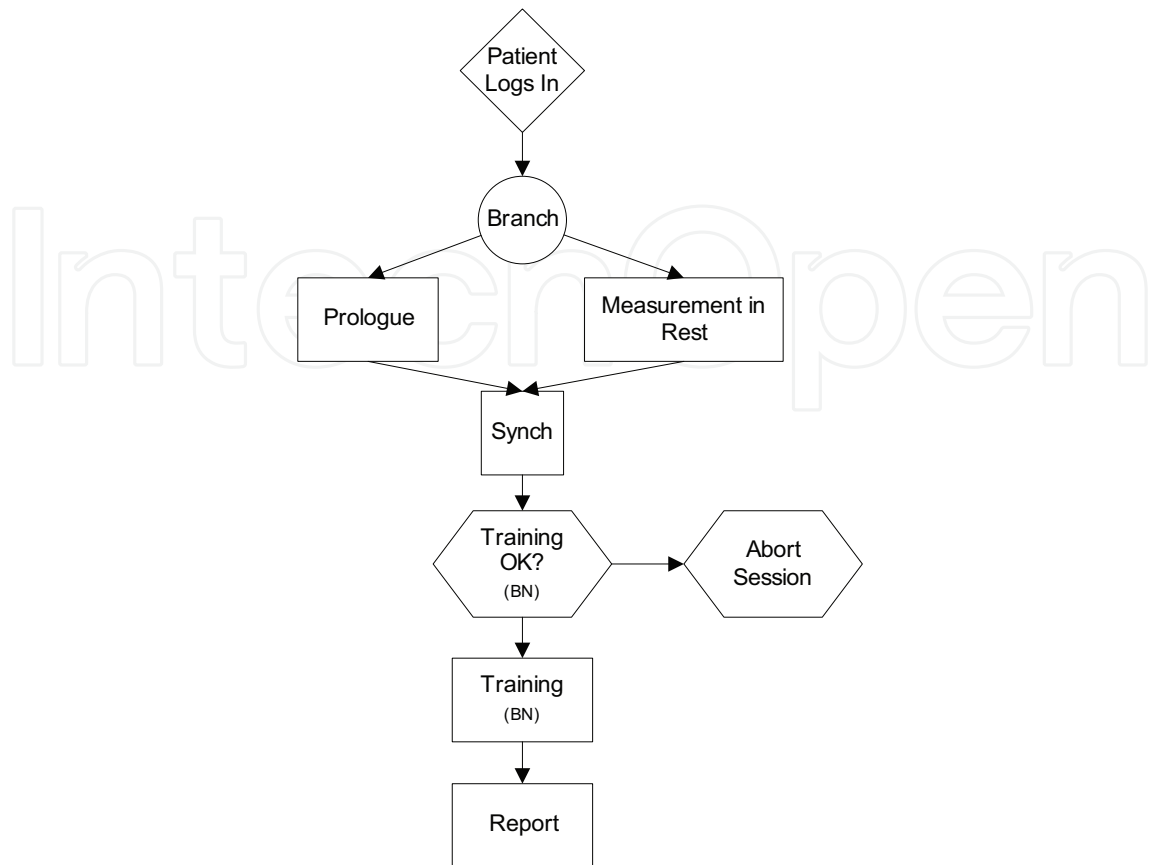


Fig. 8. Training control with a GLIF guideline

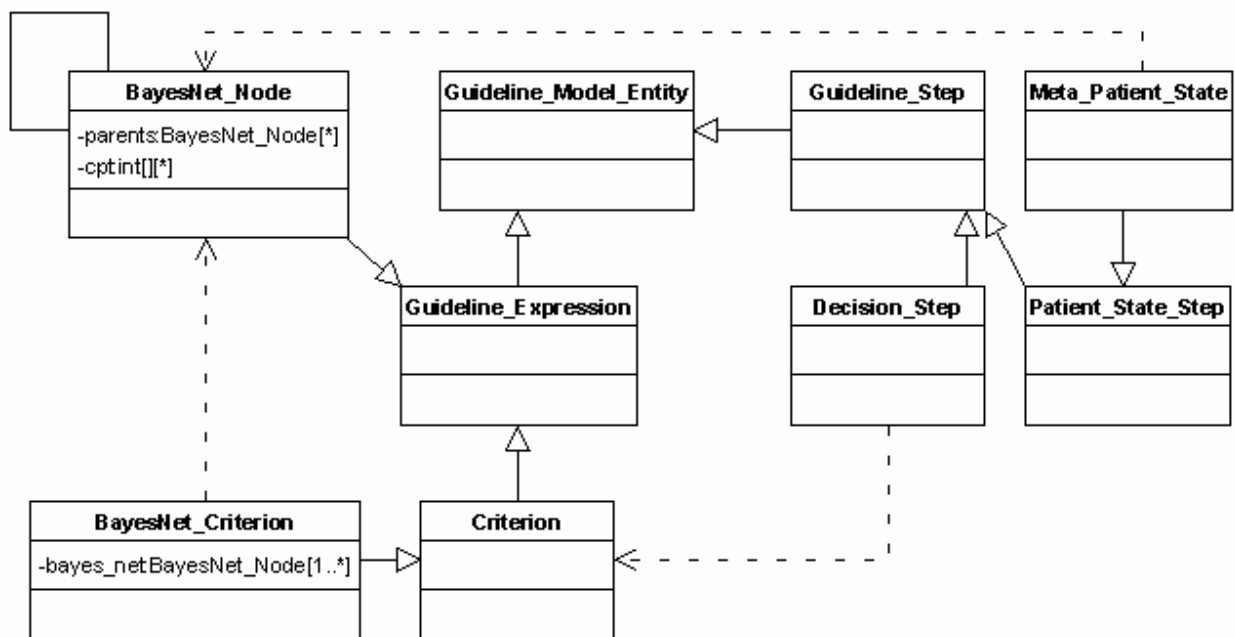


Fig. 9. Extended Guideline Interchange Format

extensions to specify a Bayesian network directly in GLIF using the Protégé editor, this is not the preferred method because of the poor support for tables in the underlying OWL model. This is why the Bayesian networks have been created using a third-party application, with the BayesianNetwork class merely pointing to the file created by the application. That application is also used to fine-tune the CPTs and to evaluate the Bayesian network based on the findings entered during the guideline execution process.

#### 5.4 Evaluation

The approach was evaluated by constructing a Bayesian network that realizes the decision processes that take place in the step 'Training OK?' shown in Fig. 8. A slightly simplified version of that network is shown in Fig. 10.

The topology of the network is largely defined by the availability of data items. In order to compare this approach with the rule-based approach described in section 4.2, no additional data should be used for the decision making process. The node labelled 'Alert' is not a chance node but a deterministic node that yields 'true' when any of the sensor alert nodes is true. The sensor alert nodes are simple Boolean chance nodes. In order to map the results to the traffic light schema of the rule-based approach, the sensor data of the sensor that triggered the alert would need to be examined for risk stratification.

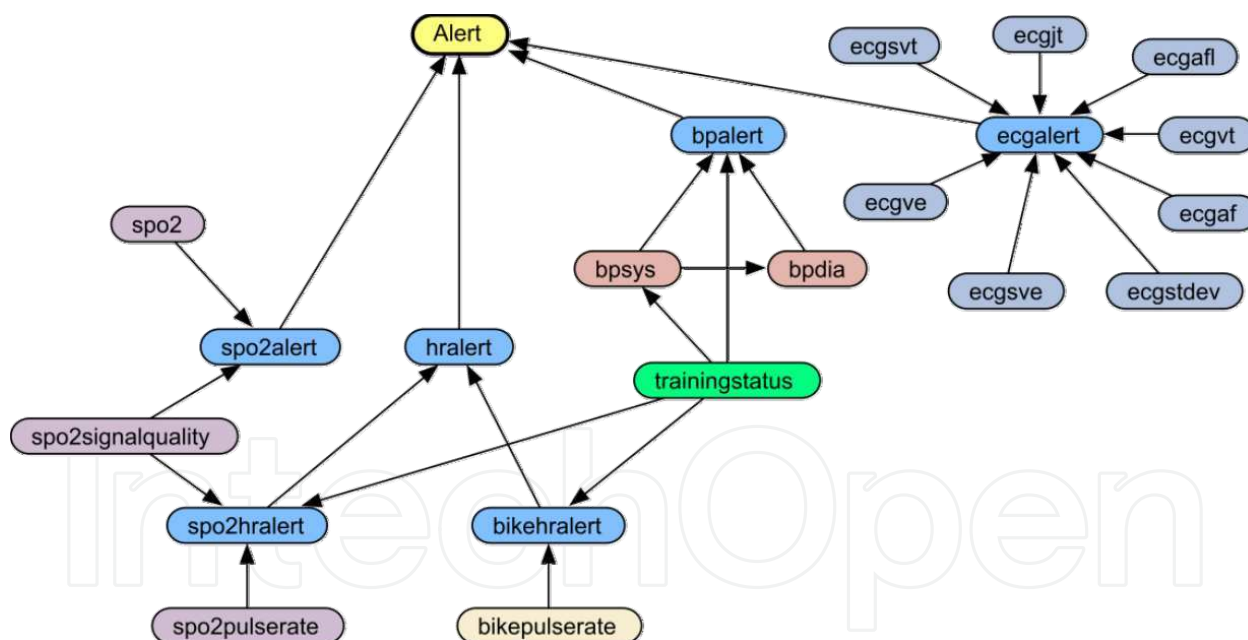


Fig. 10. Bayesian network for alert generation

The reason the alert system has been reduced to Boolean values lies in the number of states of the random variables and the conditional probability tables (CPTs) defined over these states. As mentioned in the introduction of section 5, CPTs can be specified by domain experts, be learned through data or derived through a combination of the two approaches. In the evaluation, we wanted to build the necessary CPTs by using a number of reference sessions, for which the sensor data and the results of the rule-based alarm system had been



collected. In order to be tractable, the number of states of each random variable needs to be smaller than the range of sensor values. For example for the node representing the systolic blood pressure measured in mmHg has six states, sorting the measurements into 'buckets' representing values from smaller than 100 mmHg to values greater than 160 mmHg. The size of these 'buckets' has to be small enough to represent the medical meaning but big enough to gather enough experience for the learning process. For sensors like the blood pressure sensors that deliver only a few measurements during a training session, this process is harder than for sensors like the pulse oximeter which yield one measurement per second.

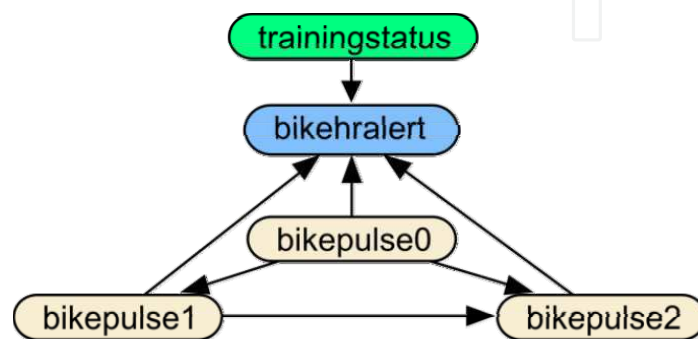


Fig. 11. Expanded portion of the alert net

As mentioned in section 4.2, alerts would be triggered too easily by motion artifacts if only the current measurement is used to decide whether or not a threshold has been exceeded. The rule-based approach of taking three measurements into consideration is used for the Bayes net version as well. This is shown in Fig. 11 for the portion that uses the pulse rate measured by the ergometer bike to determine if the patient's heart rate is too high depending on the phase (or status) of the training session. This is done for data from the pulse oximeter, the ergometer bike and for the results from the ECG analysis software. Although more than one measurement or analysis result is taken into consideration, this does not mean that no decision is made until all values are present. It merely means that the degree of belief increases with each measurement that is available.

The resulting network consists of 57 nodes, which are connected by 84 links and contain 10526 conditional probabilities. Rather than learning the CPTs for one big network, the network has been broken down into a branch dealing with ECG analysis results, a branch for the blood pressure measurements, one for the heart rate and one for the blood oxygen. In order to collect enough experience for the blood pressure nodes, 29 measurements taken during 10 completed sessions have been used. The remaining nodes were learned by using data from five typical sessions of that patient. So in addition to the blood pressure measurements, 5252 SpO<sub>2</sub> measurements, 4395 ergometer measurements, and 364 ECG analysis results were used to build the CPTs.

Once the CPTs were learned, the network was tested using sessions that had not been used for the learning process. The result was a reduction of false-positive event reports by 78%. Further testing revealed that the initial error rate of the ECG alert network of 2.496% increases greatly for previously unobserved results. This is true to an even greater degree

for the blood pressure network. However, this result is not surprising, considering how little data was available for the learning process. For example if no blood pressure of 160/100 mmHg has been observed, there is no experience that would define how the system should react. This shortcoming can easily be mitigated by entering such experience manually, allowing the network to learn and to fine-tune the CPTs accordingly. In a practical setting, the treating physician would usually validate the data and the analysis results by inspecting reports generated by the system. If the system's predictions are not as expected, the new data (including the physician's corrections) can be incorporated, automatically fine-tuning the conditional probability distributions further until the outcome meets the expectations.

## 6. Further applications

One of the advantages of the approach described in the previous section is the exploitation of the semantic layers of GLIF. Ordinarily, a Bayesian network would be agnostic about the semantic categories of the random variables and the states, making the deployment of a network in other environments a cumbersome task. However, with the extensions shown in Fig. 10 and the semantic mediation techniques demonstrated by Laleci et al. (2008), semi-automatic deployment of guidelines containing Bayesian networks is possible. This allows modifications and extensions of the scenario, adding new sensors (such as a glucose meter for diabetic patients, or a spirometer for patients suffering from lung problems). The alarm system discussed in section 5 is only one possible application of a Bayesian network. As mentioned in section 5.1, these networks can be used for other aspects of decision making as well. With influence diagrams, the utility of decisions can be considered when proposing alternative options. This would typically be exploited in the decision support system that helps the physician plan the patient's therapy. Embedding Bayesian networks in GLIF combines the strengths of GLIF on a macro level and the power of Bayesian networks for diagnostic and prognostic reasoning and for the discovery of functional interactions.

The scenario that motivated this work is from the medical field, but the applications of are not limited to it. After all, the problem of making decisions under uncertainty with incomplete and potentially unreliable data is not one that is unique. In this context, GLIF should be seen as framework for modelling workflows. To make this approach more suitable for other fields, GLIF could be replaced by other modelling workflow formalism, such as the XML Process Definition Language (XPDL).

## 7. Conclusion

In this chapter the state of the art of decision support in medicine along with clinical guidelines has been described. Especially in fields of medicine where workflows are highly standardized and do not depend in a great extend on intra-individual variations, clinical guidelines are well accepted. In these fields the clinician can benefit from computer assistance. The implementation of a CDSS requires on one hand the formalization of the clinical guidelines and mechanisms for reasoning. This chapter is focused on GLIF for the formalization and a rule-based approach for the execution of the guideline. As a potential application this approach has been used for the monitoring of a patient at home during his/her rehabilitation training. The patient benefit from the CDSS because a clinician cannot

continuously monitor the training and in case of the breakdown of the connection to the clinic the systems has to act autonomously. A first evaluation has shown that the rule-based approach results in a great number of false alarms. Therefore, an extension of the CDSS by a Bayesian networks has been proposed. In this way, the state of the patient can be monitored and interpreted in a more reliable manner. Especially in the context of the advent of mobile, personal and continuous worn sensors for vital parameters (pulse, ECG, blood pressure, SpO<sub>2</sub>, etc.) Bayesian networks for the modelling of the patient's state seem to be a good approach to provide reliable and high-quality data at discrete time points where a diagnostic or therapeutic workflow and decision process requires these data.

## 8. References

- Andreassen, S. et al. (1987). MUNIN - A causal probabilistic network for interpretation of electromyographic findings. *Proceedings of the 10th International Joint Conference on Artificial Intelligence*, pp. 366-372, Milan (Italy), August 1987, Morgan Kaufman, San Francisco
- Barnett, O. et al. (1987). DXplain. An evolving diagnostic decision-support system. *Journal of the American Medical Association*, Vol. 258, No. 1, pp. 67-74,
- Beinlich, I. et al. (1989). The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. *Proceedings of the Second European Conf. on Artificial Intelligence in Medicine*, pp. 247-256, Springer, Berlin
- Berner, E. et al. (1999). Effects of a Decision Support System on Physicians' Diagnostic Performance. *Journal of the American Medical Informatics Association*, Vol. 6, No. 5, pp. 420-427. ISSN 1067-5027
- Bertrand, M. et al. (2004). Management of acute coronary syndromes in patients presenting without persistent ST-segment elevation. *European Heart Journal*, 2004, Vol. 23, No. 23, pp. 1809-1840, ISSN 0195-668x
- de Clercq, P. et al. (2004). Approaches for creating computer-interpretable guidelines that facilitate decision support. *Artificial Intelligence in Medicine*, Vol. 31, No. 1, pp. 1-27, ISSN 0933-3657
- Elstein, A. (2004). On the origins and development of evidence-based medicine and medical decision making. *Inflammation Research*, Vol. 52, Suppl.2 , pp. 184-189, ISSN 1023-3830
- Field, M. & Lohr, K. (eds.). (1994). *Guidelines for Clinical Practice: From Development to Use*, National Academies Press, ISBN 0-30904589-4
- Field, T. et al. (2009). Computerized Clinical Decision Support During Medication Ordering for Long-term Care Residents with Renal Insufficiency. *Journal of the American Medical Informatics Association*, Vol. 16, No. 4, ISSN 1067-5027
- Galan, S. et al. (2002). NasoNet, modeling the spread of nasopharyngeal cancer with networks of probabilistic events in discrete time. *Artificial Intelligence in Medicine*, Vol. 25, No. 3, pp. 247-264, ISSN 0933-3657
- Goud, R. et al. (2005). A computer-based guideline implementation system for cardiac rehabilitation screening. *Proceedings of Computers in Cardiology*, pp. 323-326, ISBN 0-78039337-6, Lyon, September 2005, IEEE Computer Society Press

- Heckerman, D. E.; Horvitz, E.J. & Nathwani, B.N. (1992). Toward normative expert systems: Part I - The Pathfinder Project. *Methods of Information in Medicine*, Vol. 31, No. 2, pp. 90-105, ISSN 0026-1270
- Isern, D. & Moreno, A. (2008). Computer-based execution of clinical guidelines: A review. *International Journal of Medical Informatics*, Vol. 77, No. 12, pp. 787-808, ISSN 1386-5056
- Laleci, G.B. et al. (2008). SAPHIRE: A Multi-Agent System for Remote Healthcare Monitoring through Computerized Clinical Guidelines. In: *Agent Technology and e-Health*, Annicchiarico, R.; Cortés, U. & Urdiales, C. (eds.), pp. 25-44, Birkhäuser Basel
- Leibovici, L. et al. (2007). The TREAT project: decision support and prediction using causal probabilistic networks. *International Journal of Antimicrobial Agents*, Vol. 30, Suppl. 1, November 2007, pp. 93-102, ISSN 0924-8579
- Lucas P. J. F.; van der Gaag, L. C. & Abu-Hanna, A. (2004). Bayesian networks in biomedicine and health-care. *Artificial Intelligence in Medicine*, Vol. 30, No. 3, pp. 201-214, ISSN 0933-3657
- Luciani, D.; Marchesi, M. & Bertolini, G. (2003). The role of Bayesian Networks in the diagnosis of pulmonary embolism. *Journal of Thrombosis and Haemostasis*, Vol. 1, No. 4, pp. 698-707, ISSN 1538-7933
- Owens, D. et al. (1997). Representation and Analysis of Medical Decision Problems with Influence Diagrams. *Medical Decision Making*, Vol. 17, No. 3, pp. 241-262, ISSN 0272-989X
- Neapolitan, R. (2003). *Learning Bayesian Networks*, Prentice Hall, ISBN 0-13-012534-2, NJ
- Nee, O. et al. (2008). SAPHIRE: Intelligent Healthcare Monitoring based on Semantic Interoperability Platform - Pilot Applications, *IEE Proceedings*, Vol. 2, No. 2, February 2008, pp. 192-201
- Ohno-Machado, L. et al. (1998). The GuideLine Interchange Format: A Model for Representing Guidelines. *Journal of the American Medical Informatics Association*, Vol. 5, No. 5, pp. 357-372, ISSN 1067-5027
- Paul, M. et al. (2006). Improving empirical antibiotic treatment using TREAT, a computerized decision support system: cluster randomized trial. *Journal of Antimicrobial Chemotherapy*, Vol. 58, No. 6, pp. 1238-1245, ISSN 0305-7453
- Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufman Publishing, ISBN 1-55860479-0
- Peleg, M. et al. (2003). Comparing Computer-interpretable Guideline Models: A Case-study Approach. *Journal of the American Medical Informatics Association*, Vol. 10, No. 1, pp. 52-68, ISSN 1067-5027
- Seebold, H.; Lüdtke A. & Möbus, C. (2005). Bayesian Belief Network based Diagnostics in a Problem-oriented Learning Environment for Cardiology. *Proceedings of Training, Education Simulation International*, Maastricht, March 2005, Nexus Media

- Sonnenberg, F.A. & Hagerty, C.G. (2006). Computer-Interpretable Clinical Practice Guidelines: Where are We and where Are We Going? In: *IMIA Yearbook of Medical Informatics*, Haux, R. & Kulikowski, C. (eds.), pp. 72-80
- Wyatt, J. & Spiegelhalter, D. (1991). Field trials of medical decision-aids: potential problems and solutions, *Proceedings of the 15th Symposium on Computer Applications in Medical Care*, pp. 3-8, Washington, 1991

IntechOpen

IntechOpen



## **Decision Support Systems Advances in**

Edited by Ger Devlin

ISBN 978-953-307-069-8

Hard cover, 342 pages

**Publisher** InTech

**Published online** 01, March, 2010

**Published in print edition** March, 2010

This book by In-Tech publishing helps the reader understand the power of informed decision making by covering a broad range of DSS (Decision Support Systems) applications in the fields of medical, environmental, transport and business. The expertise of the chapter writers spans an equally extensive spectrum of researchers from around the globe including universities in Canada, Mexico, Brazil and the United States, to institutes and universities in Italy, Germany, Poland, France, United Kingdom, Romania, Turkey and Ireland to as far east as Malaysia and Singapore and as far north as Finland. Decision Support Systems are not a new technology but they have evolved and developed with the ever demanding necessity to analyse a large number of options for decision makers (DM) for specific situations, where there is an increasing level of uncertainty about the problem at hand and where there is a high impact relative to the correct decisions to be made. DSS's offer decision makers a more stable solution to solving the semi-structured and unstructured problem. This is exactly what the reader will see in this book.

### **How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Oliver Nee and Andreas Hein (2010). Clinical Decision Support with Guidelines and Bayesian Networks, Decision Support Systems Advances in, Ger Devlin (Ed.), ISBN: 978-953-307-069-8, InTech, Available from: <http://www.intechopen.com/books/decision-support-systems-advances-in/clinical-decision-support-with-guidelines-and-bayesian-networks>

**INTECH**  
open science | open minds

### **InTech Europe**

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166  
[www.intechopen.com](http://www.intechopen.com)

### **InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen