

Second generation knowledge acquisition methods and their application to medicine

Nada Lavrač^a and Igor Mozetič^b

^aJožef Stefan Institute,
Jamova 39, 61000 Ljubljana, Slovenia
e-mail: nada.lavrac@ijs.ac.mail.yu

^bAustrian Research Institute for Artificial Intelligence,
Schottengasse 3, A-1010 Vienna, Austria
e-mail: igor@ai.univie.ac.at

Abstract

First generation expert systems rely on the use of surface knowledge, such as associational or heuristic. This knowledge is typically acquired from domain experts through exhaustive knowledge engineering sessions. On the other hand, second generation knowledge acquisition technology is characterized by two main features: the use of deep knowledge and machine learning. In the paper we review three second generation methods that partially automate the knowledge acquisition process: inductive learning of rules from examples, model-based rule learning, and qualitative model acquisition. Results of their application to some medical domains are presented. Finally, we outline different stages of expert system development. An extended expert system shell schema is presented which includes a knowledge acquisition and a knowledge explanation module.

1 Introduction

Knowledge acquisition is a field of *artificial intelligence* concerned with the development of methods, techniques and tools for building expert system knowledge bases. It has been frequently stated that the problem of knowledge acquisition is ‘the critical bottleneck’ in expert systems development [16]. Recently, the trend in knowledge acquisition has turned towards the use of automated knowledge acquisition tools based on machine learning and qualitative modeling.

A large majority of the so-called *first generation expert systems* include knowledge bases which can be characterized as *shallow* or *surface*. Surface knowledge directly states the

relation between problem specification and problem solution without referring to the underlying principles. The critical problem in expert system development is how to acquire the required body of knowledge from the experts in a form which is complete and consistent. *First generation knowledge acquisition methods* typically acquire knowledge through a process of direct articulation. There have been a number of interview techniques, knowledge acquisition methods and tools developed which can be used to facilitate knowledge base construction [39]. These methods typically use a *conceptual model* to interact with the user thus hiding the complexity and unfamiliarity of rules, semantic nets and/or frames upon which the knowledge base is actually constructed. The interaction with the user is thus conducted at the *knowledge level* by asking for a problem description in terms of the types and relationships distinct from the *symbol level* in which the knowledge is to be encoded.

Surface-level knowledge is often called *operational* since it is used in solving problems directly, without any reference to the underlying *causal* relations on which the solution is based. Also in human problem solving, experts usually already *know* the answer to a simple problem; they retrieve the solution from their ‘operational knowledge base’. However, when faced with a difficult or unusual problem, the answer can not be directly retrieved but has to be *derived* by reasoning from ‘first principles’ [47]. For that, a model of the domain is needed which states the first principles or basic rules from which operational decisions can be made. Such knowledge is usually referred to as *deep* knowledge.

The main characteristics of *second generation expert systems* are the following. First, they include also *deep* knowledge bases which (as opposed to shallow knowledge) capture the underlying principles and structure of the problem domain [47], and second, the knowledge acquisition process is at least partially automated [30]. *Second generation knowledge acquisition methods*, based on *qualitative modeling* and *machine learning*, view knowledge acquisition as a process of modeling real-world systems, i.e., a process of developing *computer models* of the problem domains. This approach is distinct from the creation of conceptual models; developing computer models is thus not necessarily done by replicating how people think, which is a subject matter of psychology [13]. Although close attention should be paid to how experts talk and what representation they use, one should not model structures in the expert’s head but rather the behavior of a real-world system.

Figure 1 shows the flow of knowledge in different knowledge acquisition paradigms. The ‘old style knowledge engineer’s route map’ [30], representing direct encoding of rules in the development of first generation expert systems, corresponds to the knowledge flow through box A. First generation knowledge acquisition methods are outlined in Section 2.

Sections 3 - 5 describe in more detail the following second generation methods for knowledge acquisition: machine learning (in particular inductive learning of rules from examples), model-based rule learning and qualitative model acquisition.

Inductive learning from expert supplied decisions is recognized as one of the successful methods for automated knowledge acquisition. A brief review of the inductive learning approaches, results of their application to medical problems and a review of some other machine learning approaches are given in Section 3. Inductive learning of rules from examples is represented by box B in Figure 1.

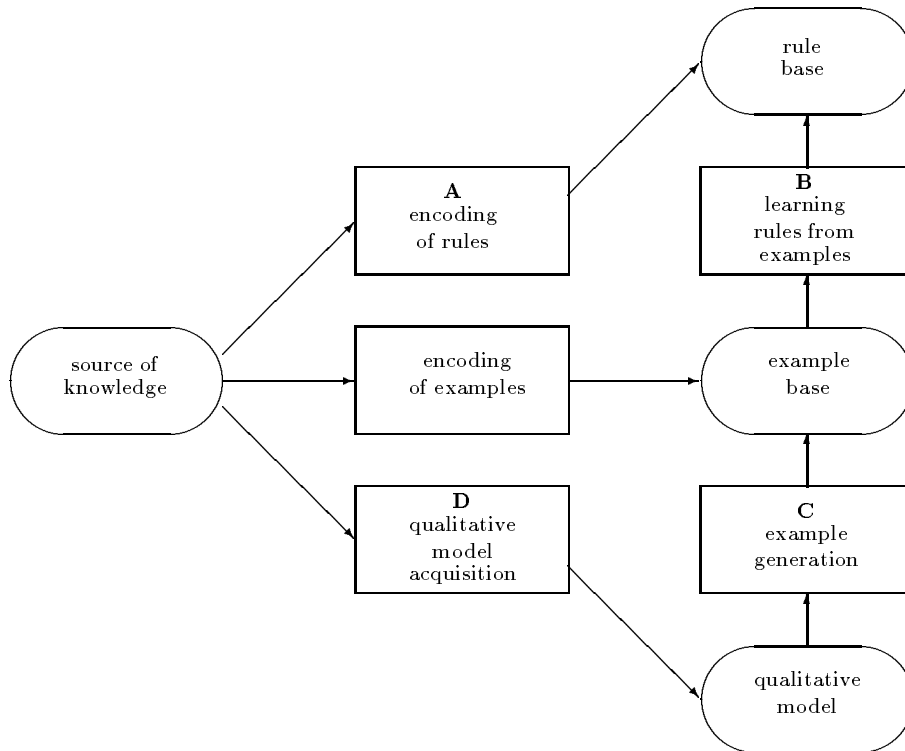


Figure 1: Knowledge flow in different methods for knowledge acquisition.

The *qualitative modeling* approach to knowledge acquisition is outlined in Section 4. Having constructed a qualitative model, one can use it to derive all possible model behaviors which in turn can be used as training examples for a machine learning system in order to induce a knowledge base of shallow-level rules. This knowledge acquisition paradigm, referred to as *model-based rule learning*, consists of two steps: example generation by a qualitative model simulation (box C in Figure 1), and learning rules from the automatically generated examples (box B in Figure 1).

The model design process can be at least partially automated by means of machine learning. An approach to semi-automatic *qualitative model acquisition* is presented in Section 5. From given partial knowledge about the system and examples of its behavior, a complete model of the system is hypothesized; the generated model is then refined by invoking an interactive debugger. This knowledge acquisition paradigm is represented by box D in Figure 1.

All the above second generation knowledge acquisition methods were used in the development of KARDIO, an expert system for the ECG diagnosis of cardiac arrhythmias [7]. The KARDIO methodology shows how the qualitative modeling approach and the machine learning technology can be used to construct knowledge bases whose complexity is far beyond the capability of traditional dialogue-based techniques for knowledge acquisition.

Tools based on the reviewed methods could be incorporated into a knowledge acquisition module of a second generation expert system shell, introduced in Section 6. Apart from presenting an extended expert system shell schema (which includes a knowledge acquisition and a knowledge explanation module), this section discusses the different stages of an expert system development in which *knowledge base refinement* has a substantial role.

2 First generation knowledge acquisition methods

In the traditional view, knowledge acquisition involves three stages:

- elicitation of data from the expert,
- interpretation of the data to infer the underlying knowledge and reasoning process; and, guided by this interpretation,
- creation of a conceptual model of the expert's domain knowledge and performance.

The first phase is usually called the *domain definition* or the *domain orientation* phase, and knowledge acquisition in this phase is usually referred to as *knowledge elicitation*. The second phase is also called *problem identification*, and the third phase is sometimes called *problem analysis* [39]. In this view, knowledge acquisition involves creating a *conceptual model* of expert knowledge and reasoning, from the analysis of data elicited by these techniques.

Many psychological and machine-aided techniques were developed aiming to solve the knowledge acquisition problem [48, 21]. *Psychological techniques* cover a range of interview strategies, observational methods, some multidimensional techniques, and verbal protocol analysis, described in more detail in a review paper by Neale [39]. These techniques, most commonly used in the domain definition phase, are used to discover what knowledge the expert uses and the methods he/she employs for problem solving within the particular domain. The analysis and structuring of this knowledge should lead to a model of the domain concepts, relations and inference strategies, corresponding to the expert's view of how such problems are solved.

Lately, the trend in knowledge acquisition seems to have turned from psychological techniques to the use of specialized tools for *machine-aided knowledge acquisition*, which can be divided into knowledge engineering languages and system-building aids [39].

Knowledge engineering languages can be further classified according to their complexity into shells and hybrid toolkits. *Shells* are simple tools, often restricted in the facilities they offer. They offer a faster and a cheaper route for expert system development, but constrain the designer with the limited formalism they offer. Most shells are restricted to one kind of reasoning, usually the backward chaining of rules. *Hybrid toolkits* offer a choice of knowledge representation and inference methods. Although they potentially offer more than shells, they don't offer any guidance as to which of the possibilities should be used under what circumstances; this is particularly a problem to non-AI programmers.

System building aids are divided into inductive tools and knowledge acquisition tools. Since inductive learning methods are, in our view, second generation knowledge acquisition methods, *inductive tools* are described in Section 3. There are a number of specialized *knowledge acquisition tools* [3], such as AQUINAS [4], KADS [10], MORE [23], DEX [1] and many others. Most of the systems use a conceptual model to interact with the user in order to hide the complexity and unfamiliarity of the underlying ‘symbol-level’ model upon which the knowledge base is actually constructed.

Many knowledge acquisition systems have specific features which distinguish them from the others. For example, a part of AQUINAS enables rapid prototyping by expressing the expert’s knowledge in the form of a rating (repertory) grid, and generates rules from this accompanied by certainty factors based on the relative strength of the rating and of the relative importance of the construct in each case. Another system DEX [1], is a specialized tool for multi-attribute decision making, also allowing for rapid knowledge acquisition. In a dialogue with the user, the system builds its knowledge base in the form of tree-structured criteria and utility functions which define the propagation of values of the criteria from the bottom to the top of the tree. In part, knowledge acquisition in DEX is based on the interactive acquisition of examples of expert decisions. As such, DEX is a step towards second generation knowledge acquisition systems.

To conclude, knowledge acquisition is typically a demanding mental process, where the knowledge engineer collaborates with domain experts. In this process the knowledge engineer’s objective is to convert human know-how into ‘say-how’ through a process of direct articulation. Such dialogue-based *direct encoding of rules* (semantic nets, frames, etc.) encounters the ‘bottleneck problem of applied artificial intelligence’ [16] and is named the ‘old style knowledge engineer’s route map’ by Michie [30]. In Figure 1, which shows the flow of knowledge in different knowledge acquisition paradigms, this ‘route map’ corresponds to the knowledge flow through box A.

3 Machine learning

Machine learning aims at automating the knowledge acquisition process to the greatest possible extent. Therefore, as opposed to some other authors [39], we consider machine learning methods to be second generation knowledge acquisition methods. Their categorization into first generation methods might be based on the fact that, traditionally, machine learning methods could only be used to acquire shallow operational knowledge. However, the emerging field of *inductive logic programming* [36] provides for methods and tools for learning deep relational descriptions, including deep qualitative models [8].

This section gives a brief review of the inductive learning approaches (Section 3.1), presents an inductive learning program ASSISTANT (Section 3.2), gives results of its application to three medical problems (Section 3.3) and provides a brief review of some other machine learning approaches to knowledge acquisition (Section 3.4).

3.1 Inductive learning of rules from examples

Inductive learning technology can be used to construct expert knowledge bases more effectively than traditional dialogue-based techniques for knowledge acquisition. As such, it can be used to generate new human-type knowledge (concept descriptions, domain models, theories, rules of expert behavior) from stored data typically captured from real-world measurements, for example in medical, engineering or scientific databases.

The method of *learning rules from examples* is recognized by Michie [30] as a ‘new style knowledge engineer’s route map’ where rules are elicited from the experts to the machine memory via the language of examples rather than via explicit articulation. Effective algorithms for inductive inference are required. There are a number of inductive learning programs such as programs of the TDIDT family (Top-Down Induction of Decision Trees) [43] or the AQ family [29] that accept tutorial examples and induce knowledge in the form of decision trees or rules, respectively. In Figure 1, this process is represented by box B where the source of knowledge is either an expert formulating a series of thoroughly chosen examples or preferably an existing database of examples interpreted and ‘cleaned’ with the help of the expert.

Learning in real-life domains often encounters the problem of dealing with imperfect data. Several TDIDT learning programs contain mechanisms for dealing with noisy, incomplete and inexact domains, for example C4 [44], CART [9] and ASSISTANT [6, 12]. The main mechanism for handling imperfect data in TDIDT programs is *tree pruning*. The other type of programs which learn from imperfect data are rule induction programs, such as AQ15 [29] and CN2 [14]. In AQ15, a technique to cope with imperfect data is *rule truncation*. In tree pruning and rule truncation, the unreliable parts of the induced descriptions are eliminated in order to increase their predictive accuracy.

Inductive learning techniques have been applied to many types of medical tasks, in particular to medical diagnosis and prognosis [6, 44]. Results in inductive learning are beginning to alleviate the knowledge acquisition problem in the development of medical expert systems. In Section 3.2 the inductive learning system ASSISTANT is described; it was developed to handle erroneous and missing data which occur frequently in medical databases. Results of its application to three real-life medical problems are given in Section 3.3.

One major dimension along which one can differentiate among machine learning systems is the complexity of the concept description languages they employ. The programs of the TDIDT family induce concept descriptions in the form of decision trees, expressed in an attribute-value language. In the AQ family of programs, the induced descriptions have the form of if-then rules expressed in a specific attribute-value formalism called VL1 (Variable-valued Logic 1).

A new area in machine learning, called *inductive logic programming* [36], is concerned with the development of programs which induce relational descriptions in some restricted first-order formalism. A restricted form of logic programs is used in MIS [46], CIGOL [37], GOLEM [38], FOIL [45] and LINUS [26].

So far, to our knowledge, LINUS was the only inductive logic programming system

that addressed the problem of inducing medical diagnostic rules, in particular the problem of learning rules for early diagnosis of rheumatic diseases [27]. It was possible to apply LINUS to these problems for two main reasons. First, LINUS is able to tackle problems which combine attribute and relation learning. And second, LINUS is an environment that integrates different attribute-value learning algorithms. Since the domain of early diagnosis of rheumatic diseases involves missing and noisy data, it was possible to achieve good results by using ASSISTANT as the algorithm incorporated into the LINUS inductive logic programming environment.

3.2 Inductive learning with ASSISTANT

ASSISTANT is a descendant of ID3 [43]. The system generates a concept description in the form of a decision tree that enables classification of new objects. As such, it is a member of the TDIDT family of inductive learning programs.

ID3 implements a simple mechanism for discovering a classification rule from a training set of objects belonging to two classes. Each object is described in terms of a fixed collection of attributes, each of which has its own set of values. The system builds a classification rule in the form of a decision tree which correctly classifies all the given objects. Each of the interior nodes of the tree is labeled by an attribute, while branches that lead from the node are labeled by its possible values. The tree construction is heuristically guided by choosing the most informative attribute at each step, aimed at minimizing the expected number of tests. ID3 implements the following basic algorithm:

```
if    all training instances belong to the same class
then generate a leaf labeled with the class name
else  1. select the most informative attribute for the node
      2. split the training set into subsets according to the values of
         the selected attribute
      3. recursively repeat the procedure for each subset
```

In ASSISTANT, the basic ID3 learning algorithm was extended in several ways which include: handling problems with more than two decision classes, handling incompletely specified training examples, binarization of continuous attributes, binary construction of decision trees, pruning of the unreliable parts of a tree and plausible classification based on the ‘naive’ Bayesian principle.

One of the most important features is tree pruning, used as a mechanism for handling noisy data. Tree pruning is aimed at producing smaller trees, which do not overfit possibly erroneous data. There are two types of pruning:

- *pre-pruning*, performed during the construction of a decision tree, which decides whether to stop or to continue the tree construction at a certain node, and
- *post-pruning*, used after the decision tree is constructed, which estimates the classification errors in the nodes and then decides whether to prune certain subtrees or not.

Experiments in several domains have shown that these features produce decision trees which are smaller and easier to understand, and at the same time more accurate when classifying new objects.

3.3 Applications to medical domains

ASSISTANT has been applied to a number of medical domains, among others the prognosis of breast cancer recurrence, location of primary tumor, and lymphography [6, 12, 35]. Results of the experiments are given in Table 1. In the experiments, 70% of the available examples were randomly selected for training, and the remaining 30% were used for testing the accuracy of automatically constructed diagnostic trees/rules. Experiments were repeated 4 times; given are the average results.

Prognosis of breast cancer recurrence. For about 30% of patients that undergo a breast cancer operation, the illness reappears in five years. Prognosis of this recurrence is very important for patients' post-operational treatment. The domain is characterized by 2 possible prognoses and 9 attributes (age, size and location of tumor, etc.). The data on 286 patients with known diagnostic status five years after the operation were available. The data are not complete and the set of attributes is not always sufficient for distinguishing between the two classes. Using this data, ASSISTANT induced prognostic rules which are, on average, 72% accurate. For comparison, five specialists of the Ljubljana Medical Center, Institute for Oncology, were tested using the same testing examples. They gave a correct prognosis in 64% of cases.

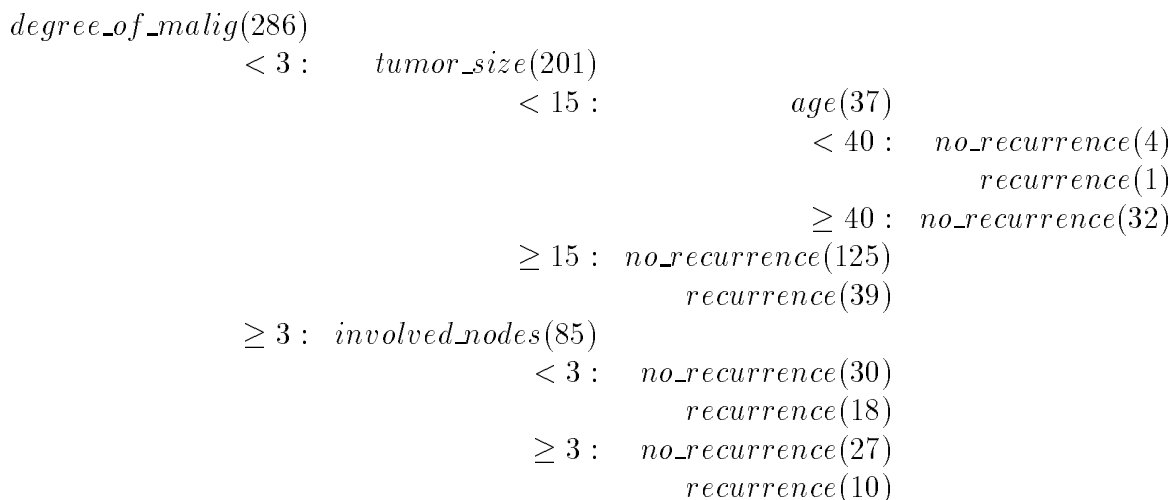


Figure 2: Decision tree for the prognosis of breast cancer recurrence, induced by ASSISTANT. Numbers associated with each node denote the number of training examples in the node.

Figure 2 gives an induced decision tree for the prognosis of breast cancer recurrence,

derived by ASSISTANT. The numbers of training examples in the leaves of the decision tree are transformed into probabilities which are then used in the classification of unseen cases. The uppermost branch of the decision tree in Figure 2 can be interpreted as follows:

if the *degree of malignancy* is less than 3
 and the *tumor size* is less than 15
 and the patient's *age* is less than 40
then there are 4 out of 5 supporting examples that the *cancer will not recur*
 and 1 supporting example that the *cancer will recur*.

Below is a description of two other medical domains, location of primary tumor and lymphography.

Location of primary tumor. Physicians distinguish between 22 locations of primary tumor. Patients' data are described by 17 attributes (age, sex, histologic type of carcinoma, degree of differentiation, and possible locations of detected metastases). The data on 339 patients with known locations of primary tumor (verified by operation or by X-ray) were available for the experiment. Again, the data are not complete; for some patients the data on the histologic type and degree of differentiation are missing. Furthermore, the given set of attributes is not always sufficient to distinguish between the different locations. ASSISTANT gave a correct result in 46% of cases, on average. At the Ljubljana Medical Center, Institute for Oncology, four internists and four specialists were asked to give their diagnosis. Internists determined a correct location of primary tumor in 32% and oncologists in 42% of cases, which indicates the difficulty of the diagnostic problem. Namely, there are 22 possible locations and the correct locations of primary tumor is only one of the evidences used in cancer treatment.

Lymphography. The domain is characterized by 18 attributes (sex, age, different data about nodes and laboratory tests) and 4 diagnoses. The data on 148 patients were available. Diagnoses in this domain were not verified and the actual testing of physicians was not performed. The specialists' estimate was that internists diagnose correctly in about 60% and specialists in about 85% of cases.

Medical domain	ASSISTANT	AQ15	Medical specialists
Lymphography	77%	82%	85% (estimate)
Breast cancer	72%	68%	64%
Primary tumor	46%	41%	42%

Table 1: Diagnostic accuracy achieved by the learning programs ASSISTANT and AQ15, averaged over 4 experiments, and compared to medical specialists.

Table 1 gives the results of applying ASSISTANT to the three medical problems. For comparison, the results of the performance of another inductive learning system AQ15 and of the medical experts are presented [29]. In the four experiments, the same testing set of

30% of examples was used both for testing the accuracy of the automatically constructed diagnostic rules and the accuracy of medical experts. Results show that the predictive accuracy of both systems is at the level of domain experts.

3.4 Some other machine learning approaches to knowledge acquisition

For the sake of completeness, we briefly outline three other machine learning approaches which can be used in knowledge acquisition: the genetic, analytic, and connectionist learning paradigm [11].

Genetic algorithms [5] have been inspired by a direct analogy to mutations in biological reproduction. A genetic algorithm selects high strength classifiers as ‘parents’, forming ‘offspring’ by recombining components from the parent classifiers. The offspring displace weak classifiers in the system and enter into competition, being activated and tested when their conditions are satisfied. Thus a genetic algorithm mimics the genetic processes underlying evolution. Genetic algorithms discover rules for *classifier systems* which are massively parallel, message passing, rule-based systems, and which operate in environments with the following characteristics: perpetually novel events accompanied by large amounts of noisy or irrelevant data; continual, often real-time, requirements for action; implicitly or inexactly defined goals; and sparse payoff of reinforcement obtainable only through long action sequences. Classifier systems are designed to absorb new information continuously from such environments, devising sets of competing hypotheses (expressed as rules) without disturbing significantly capabilities already acquired.

The *analytic learning* paradigm assumes existence of a rich domain theory, and one (or a few) examples representing past problem solving experience. Examples are used to guide the selection of the deductive chains when solving new problems, or to formulate search control rules that enable more efficient application of domain knowledge. In analytic methods, an instance corresponds to a portion of a problem solving trace, and learning uses this (single) instance and background knowledge (domain theory). Much research is devoted to *explanation-based generalization* methods [31]. The result of using these methods is nothing but *operational* surface rules, which could (at least in principle) be deduced from the given domain theory. Since the analytic learning methods are deductive, this type of learning performs at the symbol level and not at the knowledge level, i.e., deductive closure of the domain theory does not increase.

Connectionist learning systems [22], also called *neural networks* or *parallel distributed systems*, have lately gained much attention. Connectionist models typically consist of many simple, neuron-like processing units that interact using weighted connections. Each neuron has a state (activity level) that is determined by the input received from other neurons in the network. A neural network learns to discriminate between classes, given a set of instances of each class, in a holistic manner. Learning consists of readjusting weights in a fixed-topology network via different learning algorithms.

A *Bayesian neural network* [24] is an efficient tool for learning classification knowledge.

In this type of network each neuron implements the ‘naive’ Bayesian classifier, therefore one iteration in the network corresponds to one classification with ‘naive’ Bayesian rule (where ‘naive’ stands for the independence assumption of events). Iterations correct noisy data and approximate missing data. An expert system based on the Bayesian neural network does not contain a knowledge base in the classical sense. Each neuron represents an event. The knowledge is stored in the weights associated to connections between neurons. Each such weight can be directly interpreted since it represents the probability that two events occurred at the same time. The approach is similar to the one used in Inferno [42], having the same structure of the network. In Bayesian neural networks, the generalization produces *fixed* points which correspond to induced if-then rules; inductive learning systems tend to generate few general rules, while in the network there are many specialized fixed points.

4 Model-based rule learning

Qualitative modeling is concerned with the development of *deep* knowledge bases that capture the underlying causal structure of the problem domain. Such knowledge can be represented in the form of a model that states the first principles or the basic ‘rules of the game’ from which operational decisions can be derived. The prevailing type of knowledge in such a model is *qualitative* [15, 18, 25, 7].

Qualitative modeling has several advantages over the conventional numerical modeling: the qualitative view is often closer to human reasoning about the physical or physiological processes being modeled; to execute the model one does not have to know the exact numerical values of the parameters in the model; a qualitative simulation may be computationally less complex than numerical simulation. Qualitative simulation can often be used for constructing explanations of the mechanisms of a system being modeled more naturally than numerical modeling.

In principle, a qualitative model can be used for problem solving directly. It can be used to answer prediction, diagnostic and control type of questions. The *prediction* task is to find the observable results of applying some input to the system, given a functional state of the system. The *diagnostic* task is: given the inputs to the system and the observed manifestations, find the system’s functional state (normal or faulty, which components are failed). The *control* task is to determine the input control to the system, assuming its state, in order to achieve a desired output.

Since a model is usually designed for simulation and prediction, using it to solve diagnostic and control tasks can be computationally expensive. Nevertheless, by qualitative simulation, the model can be used to automatically generate examples of the possible behaviors. From such a set of examples, operational decision rules can be generated by inductive learning methods. This idea is the basis of the *model-based rule learning* knowledge acquisition paradigm, which consists of two steps: example generation by simulations of a qualitative model (box C in Figure 1) and learning rules from the automatically generated examples (box B in Figure 1).

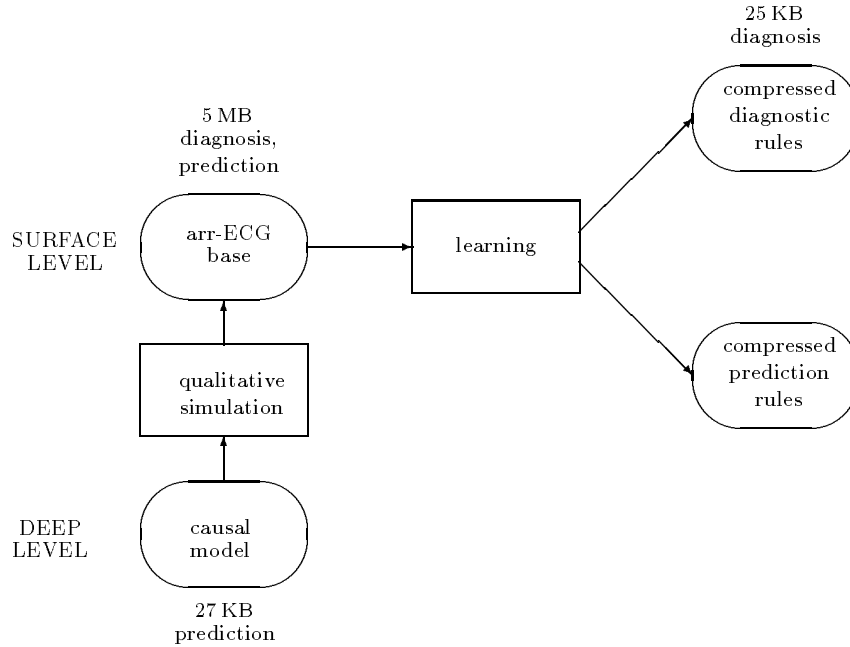


Figure 3: Deep and surface levels of cardiological knowledge, and transformations between representations.

The model-based rule learning paradigm was used in KARDIO to generate compressed diagnostic and prediction rules as shown in Figure 3. A deep qualitative model of the electrical activity of the heart was developed and used for the automatic synthesis (through simulation) of the surface knowledge about ECG interpretation. The ECG interpretation knowledge has the form of pairs $(Combined_arrhythmia, ECG_description)$ relating each of the 2,419 possible combined arrhythmias to the corresponding ECG patterns (there are altogether 140,966 ECG patterns). The surface representation facilitates fast ECG diagnosis, but it is rather complex in terms of memory space (over 5 Mb, stored as text file). This motivated the compression of the surface knowledge by means of an inductive learning program of the AQ family into a compact and diagnostically efficient representation. In Figure 3, the representations are arranged as to emphasize the distinction between the deep and surface levels of knowledge. The two steps of model-based rule learning are referred to as *qualitative simulation* and *learning*, respectively.

The *learning* step is based on the following idea: use the compiled representation (pairs arrhythmia-ECG) as a source of examples and apply an inductive learning algorithm to obtain their compact descriptions. The inductive learning program used was NEWGEM [32], a member of the AQ family.

Figure 3 shows the compression effects achieved in terms of storage space needed when storing different representations simply as text files. The figure 25 KB associated with the corresponding representations includes both the induced descriptions plus the additional

rules needed to attain logical equivalence with the exhaustive arrhythmia-ECG representation. Notice the compression factor of about 200. The similarity in size between the deep model and the compressed representations is incidental.

What follows is a prediction rule for AV block 3, generated by NEWGEM, and two corresponding descriptions from the medical literature.

```

if      AV_conduct = avb3
then    Rhythm_QRS = regular &
          Relation_P_QRS = independent_P_QRS

```

Description of the arrhythmia AV block 3 by Goldman [20]: In this condition the atria and ventricles beat entirely independently of one another. The ventricular rhythm is usually quite regular but at much slower rate (20-60).

Description of AV block 3 by Phibbs [41]: 1. The atrial and ventricular rates are different: the atrial rate is faster; the ventricular rate is slow and regular. 2. There is no consistent relation between P waves and QRS complexes.

Below is an example of a synthesized diagnostic rule:

```

if      Relation_P_QRS = after_P_some_QRS_miss
then    AV_conduct = wen  $\vee$  mob2
           $\vee$ 
          Atr_focus = afl  $\vee$  af &
          AV_conduct = normal

```

This rule corresponds to a particular diagnostic feature in the ECG, characterized by some P waves not followed (as normally) by the corresponding QRS complexes. The rule states that this feature is indicative of the defects called Wenckebach or Mobitz 2, or, when the AV conductance is normal, of the atrial flutter or fibrillation. The rule thus clearly indicates what kinds of disorders a diagnostic system should be looking for in the case that this abnormality is detected in the ECG.

The individual steps of the KARDIO methodology are described in more detail elsewhere in this volume [34]. The methodology shows how the qualitative modeling approach and the machine learning technology can be used in the development of practical expert systems. The KARDIO knowledge acquisition paradigm has already been used in the development of a satellite power supply fault diagnosis system by Pearce [40] and Feng [17].

5 Qualitative model acquisition

The model design process can be at least partially automated by means of machine learning. The Qualitative Model Acquisition System (QuMAS) [33] supports the construction of a deep model and the representation of a model at different levels of detail. In QuMAS, partial knowledge about the model and examples of its behavior are provided by the user,

and the complete model is automatically constructed and incrementally refined until the desired behavior is achieved. This knowledge acquisition paradigm is represented by box D in Figure 1.

In this approach, we restrict ourselves to functional qualitative models, where a model is defined by its structure (a set of components and their connections) and functions of the individual components. As outlined in Figure 4a, QuMAS consists of three subsystems: a *learner* that hypothesizes functions of components from examples of their behavior, an *interpreter* that can use the hypothesized model to derive its behavior, and a *debugger* that locates faulty functions of components and proposes how to correct them.

It is assumed that only partial knowledge about the model is given - its structure. Further, examples of the behavior of the model and its constituent components are provided from which the learning part of the system hypothesizes functions of the components. The interpreter of the model is then able to derive its behavior. The user can test the model and compare it with the intended behavior. When a difference between the derived and the intended behavior of the model occurs, a debugger is invoked. The debugger locates faulty hypotheses defining functions of components, proposes examples of behavior that guarantee the intended behavior of the model, and invokes the learner that incrementally refines the hypotheses.

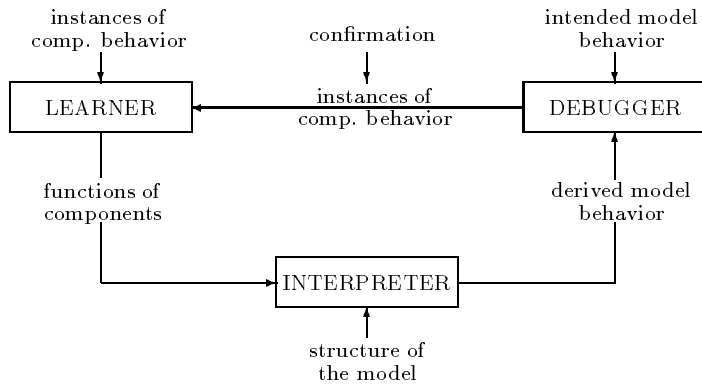
The cycle of deriving the behavior of the model, debugging the model, and incremental learning is repeated until the intended behavior of the model is achieved, i.e., until the user believes that the model is correct and complete with respect to the actual system being modeled.

QuMAS embodies two types of learning. Initial data-driven learning generalizes examples of components' behavior into rules on the basis of similarities and differences and does not require any user interaction. The second type of learning is model-driven where the debugger actively constructs examples of components' behavior which satisfy the intended model behavior, and then queries the user for confirmation. QuMAS therefore offers a trade-off between the initial amount of knowledge provided by the user, and the time one is willing to spend on debugging the model.

QuMAS is used interactively by the model designer, and takes advantage of the hierarchical model representation to speed up the automatic learning of the model (Figure 4b). The hierarchy has also a role in generating a good and concise explanation on points selected by the user. A substantial part of the KARDIO heart model was reconstructed semi-automatically using QuMAS [33, 7].

6 Refinement cycles in knowledge acquisition

Development of an expert system is essentially an iterative process which typically needs several *refinement cycles*. In each cycle, the expert and the knowledge engineer refine the knowledge base by comparing the human performance with the machine performance and the original human knowledge with the generated machine representation. The refinement cycles need to be repeated until the intended performance of the system is achieved.



(a)



(b)

Figure 4: An overview of the qualitative model acquisition system (a), and the top-down model construction method (b).

Expert system shells have to integrate a variety of tools that allow for acquisition, explanation and utilization of complex domain knowledge. The classical expert system schema shown in Figure 5a cannot cover all the required functions of the system. In this schema, an expert system consists of a domain dependent knowledge base and of a domain independent expert system shell which incorporates an inference engine and a user interface. In Figure 5b, a new schema of an expert system shell is proposed [2]. The structure of an expert system shell is extended to incorporate: the *acquisition module* that provides tools for acquiring and editing the knowledge base, the *explanation module* providing different representations of the knowledge base, and the *reasoning module* which uses the knowledge base to find solutions to a problem; it also has to provide explanations of individual solutions, and enable the knowledge base validation.

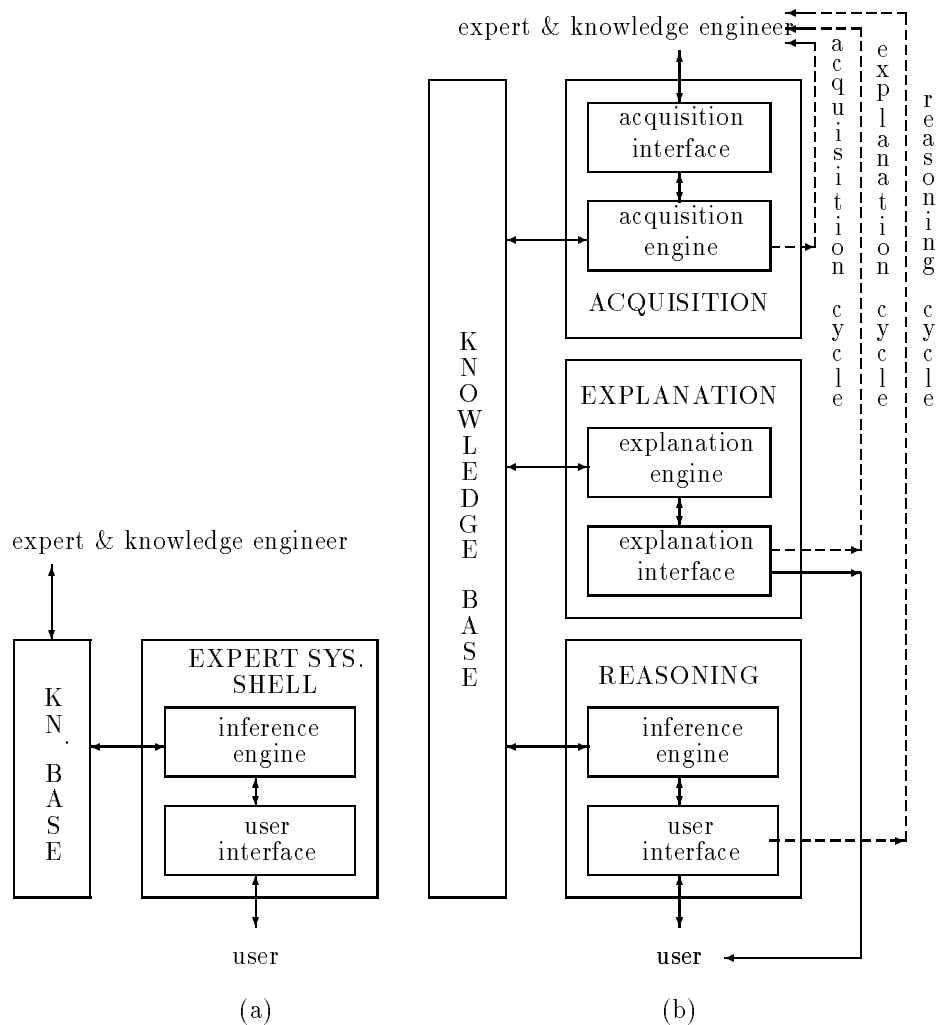


Figure 5: A classical (a) and an extended schema (b) of an expert system shell.

This new schema covers all the functional requirements of expert systems, namely

apply, explain, acquire, display, edit, and validate the knowledge base [19]. In our view, an expert system shell is not aimed just to support problem solving, but should also actively support knowledge acquisition and refinement of the elicited knowledge. Therefore, two new modules are introduced: a knowledge acquisition and a knowledge explanation module.

The knowledge acquisition module is to support one or more methods for knowledge acquisition. For example, it can incorporate a machine learning system and/or a system which supports direct or semi-automatic construction of qualitative models. The knowledge explanation module is aimed at generating different explanations - from displaying the knowledge base in a compact and comprehensible form, to representing the knowledge from different viewpoints and at different level of detail [2]. By a ‘different viewpoint’ we mean the representation of the same knowledge in a different language (e.g., graphical or tabular representation), the reorganization of knowledge (e.g., grouping rules, or expressing one set of rules with another), or the representation of additional information derived from the original knowledge base (e.g., different statistics, Bayesian probabilities, informativity of attributes, etc.). By a ‘different level of detail’ we mean knowledge representation by using attributes at different levels of a taxonomic hierarchy, or by eliminating too specific knowledge, e.g., presenting only the most important rules. Some of these features are also provided in the machine learning system ASSISTANT, described in Section 3.2.

According to our expert system shell schema, we distinguish between the *acquisition*, *reasoning* and *explanation refinement cycle* (represented by dashed lines in Figure 5b). These refinement cycles have to be supported by appropriate development tools, incorporated into the corresponding modules of an expert system shell.

Depending on the type of the system’s knowledge base, the *acquisition refinement cycle* in Figure 5b consists of a feedback loop back to the source of knowledge. In Figure 1, which illustrates the knowledge flow in different methods for knowledge acquisition, the acquisition refinement cycle could be introduced by feedback loops from any of the three types of the acquired knowledge (rule base, example base, qualitative model) back to the source of knowledge. In the case that a knowledge base is induced from a set of tutorial examples, this corresponds to Michie’s ‘first refinement cycle’ of his ‘new style knowledge engineer’s route map’ [30]. His ‘second refinement cycle’ corresponds to what we call here the *reasoning refinement cycle* in which the system’s performance is compared with the human expert’s performance.

The knowledge explanation module is aimed at providing different representations of the knowledge base to be shown to the user. This motivates the expert and the knowledge engineer to further check and elaborate the knowledge base as new ideas are triggered that may lead to the identification of inadequate or missing knowledge. In the process of knowledge acquisition we call this the *explanation refinement cycle*. In this cycle, the generated machine representation of knowledge is compared with the original expert’s knowledge. Differences between the two can result from either an error in the machine representation or a slip in the original human codification of knowledge. In the former case, the error is a consequence of incorrectly encoded rules, errors in learning examples, or an error in the deep causal model, depending on the type of the system’s knowledge base. In the latter case, the refinement cycles will expose blemishes in the existing, original

expert formulations, and may thus help to improve those.

In the man-machine dialogue through several refinement cycles both man and machine learn. We can consider the process of knowledge elicitation as a man-machine learning process consisting of the stepwise acquisition and refinement of knowledge. The level of human and machine knowledge grows. From this point of view, the role of an expert system is not only in solving problems and explaining the solutions, but also in building (new) human-type knowledge through refinement cycles. By acquiring this refined knowledge, machine knowledge is improved and allows for better performance. This man-machine learning process extends the role of expert systems to *expert support systems* [28]. In contrast to first generation expert systems whose main goal is to simulate human experts' performance in problem solving, expert support systems stimulate human mental processes in the process of knowledge acquisition, therefore supporting human learning as well.

7 Conclusion

The development of an expert system is typically an iterative process. It consists of refinement cycles that are repeated until the intended performance of the system is achieved. We emphasize the importance of the knowledge acquisition and knowledge explanation cycle that allow us to see the development of an expert system as a process in which both man and machine learn and in which (new) human-type knowledge is being generated.

The paper gives an overview of some second generation knowledge acquisition methods. Three methods are described together with the results of their application to medical problems: learning rules from examples, model-based rule learning and semi-automatic model acquisition. All these methods are aimed at automating the knowledge acquisition process.

An important advantage of using the reviewed methods, based on machine learning and qualitative modeling, is that the knowledge bases can be guaranteed to be complete and consistent with regard to the given set of training examples (in exact and non-noisy domains) and with regard to the selected level of detail of a qualitative model, respectively. In non-exact and noisy domains, however, the appropriate noise-handling mechanisms of advanced inductive learning tools can be used to deal with non-exact and erroneous data. The acquired knowledge bases can be tested for their accuracy when classifying unseen cases. Having tested the classification accuracy of the knowledge base and having estimated the statistical significance of the results, the user can statistically justify and explain the answers obtained by using an expert system containing the automatically generated knowledge base.

The reviewed methods for knowledge acquisition and refinement, all of which are used in KARDIO to model a substantial real-life problem, emphasize the role of machine learning and qualitative modeling techniques in the development of second generation expert systems.

Acknowledgements

All the described methods for knowledge acquisition were used (and two of them developed) when building KARDIO, a medical expert system for the ECG diagnosis of cardiac arrhythmias, in collaboration with Ivan Bratko. The extended expert system shell schema was proposed in collaboration with Marko Bohanec and Vladislav Rajkovič. We are grateful to Donald Michie and Tanja Urbančič for valuable comments on an earlier draft of the paper. The second author also wishes to thank Robert Trappl for creating an enjoyable working environment.

The first author is supported by the Slovene Ministry of Science and Technology, while the second author is supported by the Austrian Federal Ministry of Science and Research.

References

- [1] Bohanec, M. and Rajkovič, V. (1990) DEX: An expert system shell for decision support. *Sistemica* 1 (1), 145-158. Lima, Peru: IAS.
- [2] Bohanec, M., Rajkovič, V., and Lavrač, N. (1988) Knowledge explanation in expert systems: A decision support system and machine learning view. *IASTED Intl. Conference on Expert Systems*. Geneva, Switzerland.
- [3] Boose, J.H. (1989) A survey of knowledge acquisition techniques and tools. *Knowledge Acquisition* 1 (1), 3-37. Academic Press.
- [4] Boose, J.H., Shema, D.B. and Bradshaw, J.M. (1989) Recent progress in AQUINAS: A knowledge acquisition workbench. *Knowledge Acquisition* 1 (2), 185-214. Academic Press.
- [5] Booker, L.B., Goldberg, D.E. and Holland, J.H. (1989) Classifier systems and genetic algorithms. *Artificial Intelligence* 40 (1-3), Special Issue on Machine Learning, 235-282. Amsterdam: North Holland.
- [6] Bratko, I., and Kononenko, I. (1987) Learning diagnostic rules from incomplete and noisy data. In: Phelps, B. (ed.) *AI Methods in Statistics*. London: Gower Technical Press.
- [7] Bratko, I., Mozetič, I. and Lavrač, N. (1989) KARDIO: A study in deep and qualitative knowledge for expert systems. Cambridge, MA: MIT Press.
- [8] Bratko, I., Muggleton, S. and Varšek, A. (1991) Learning qualitative models of dynamic systems. *Eighth Intl. Workshop on Machine Learning*. Evanston, IL: Morgan Kaufmann.
- [9] Breiman, L., Friedman, J.H., Olshen, R.A. and Stone, C.J. (1984) *Classification and regression trees*. Belmont: Wadsworth.

- [10] Breuker, J., and Wielinga, B. (1987) Knowledge acquisition as modeling expertise: The KADS methodology. First European Workshop on Knowledge Acquisition for Knowledge-based Systems, EKAW 87. Reading University.
- [11] Carbonell, J.G. (1989) Introduction: Paradigms for machine learning. *Artificial Intelligence* 40 (1-3), Special Issue on Machine Learning, 1-9. Amsterdam: North Holland.
- [12] Cestnik, B., Kononenko, I. and Bratko, I. (1987) ASSISTANT 86: A knowledge elicitation tool for sophisticated users. In: Bratko, I. and Lavrač, N. (eds.) *Progress in machine learning*. Wilmslow: Sigma Press.
- [13] Clancey, W.J. (1989) The knowledge level reinterpreted: Modeling how systems interact. *Machine Learning* 4 (3/4), Special Issue on Knowledge Acquisition, 285-292. Kluwer Academic Publishers.
- [14] Clark, P. and Niblett, T. (1989) The CN2 induction algorithm. *Machine Learning* 3 (4), 261-284. Kluwer Academic Publishers.
- [15] de Kleer, J., and Brown, J.S. (1984) A qualitative physics based on confluences. *Artificial Intelligence* 24 (1-3), 7-83. Amsterdam: North Holland.
- [16] Feigenbaum, E.A. (1977) *The art of artificial intelligence 1: Themes and case studies of knowledge engineering*. Pub. no. STAN-SC-77-621. Stanford University, Department of Computer Science.
- [17] Feng, C. (1991) Inducing temporal fault diagnostic rules from a qualitative model. First Intl. Inductive Logic Programming Workshop. Viana de Castelo, Portugal.
- [18] Forbus, K.D. (1984) Qualitative process theory. *Artificial Intelligence* 24 (1-3), 85-168. Amsterdam: North Holland.
- [19] Gaines, B.R. (1987) Knowledge acquisition for expert systems. First European Workshop on Knowledge Acquisition for Knowledge-based Systems, EKAW 87. Reading University.
- [20] Goldman, M.J. (1976) *Principles of clinical electrocardiography*. Los Altos: Lange Medical Publications.
- [21] Hart, A. (1986) *Knowledge acquisition for expert systems*. London: Kogan Page.
- [22] Hinton, G.E. (1989) Connectionist learning procedures. *Artificial Intelligence* 40 (1-3), Special Issue on Machine Learning, 185-234. Amsterdam: North Holland.
- [23] Kahn, G., Nowlan, S., and McDermott, J. (1985) MORE: An intelligent knowledge acquisition tool. Ninth Intl. Joint Conference on Artificial Intelligence. Los Angeles, CA: Morgan Kaufmann.

- [24] Kononenko, I. (1989) Bayesian neural networks. *Biological Cybernetics* 61, 361-370. Springer-Verlag.
- [25] Kuipers, B. (1986) Qualitative simulation. *Artificial Intelligence* 29, 289-338.
- [26] Lavrač, N., Džeroski, S. and Grobelnik, M. (1991) Learning nonrecursive definitions of relations with LINUS. Fifth European Working Session on Learning, EWSL 91. Porto, Portugal: Springer-Verlag.
- [27] Lavrač, N., Džeroski, S., Pirnat, V. and Križman, V. (1991) Learning rules for early diagnosis of rheumatic diseases. Third Scandinavian Conference on Artificial Intelligence, SCAI'91. Roskilde, Denmark: IOS Press.
- [28] Luconi, F.L., Malone T.W., and Scott Morton, M.S. (1986) Expert Systems: The next challenge for managers. *Sloan Management Review*.
- [29] Michalski, R.S., Mozetič, I., Hong, J. and Lavrač, N. (1986) The multi-purpose incremental learning system AQ15 and its testing application to three medical domains. Fifth National Conference on Artificial Intelligence, AAAI-86, 1041-1045. Philadelphia, PA: Morgan Kaufmann.
- [30] Michie, D. (1986) Machine learning and knowledge acquisition. In: Michie, D., and Bratko, I. (eds.) *Expert systems: Automating knowledge acquisition*. Addison-Wesley. Also published as: *New developments in expert systems* In: Quinlan, J.R. (ed.) *Applications of Expert Systems*. Addison-Wesley, 1987.
- [31] Mitchell, T.M., Keller, R.M. and Kedar-Cabelli, S.T. (1986) Explanation-based generalization: A unifying view. *Machine Learning* 1 (1), 47-80. Kluwer Academic Publishers.
- [32] Mozetič, I. (1985) NEWGEM: Program for learning from examples - Technical documentation and user's guide. Technical Report, University of Illinois at Urbana-Champaign, Department of Computer Science. Also: Technical Report IJS-DP-4390, Jožef Stefan Institute, Ljubljana, Yugoslavia.
- [33] Mozetič, I. (1987) The role of abstractions in learning qualitative models. Fourth Intl. Machine Learning Workshop, 242-255. Irvine, CA: Morgan Kaufmann.
- [34] Mozetič, I. (1990) Diagnostic efficiency of deep and surface knowledge in KARDIO. *Artificial Intelligence in Medicine* 2 (2), 67-83. Also in this volume.
- [35] Mozetič, I, Lavrač, N. and Kononenko, I. (1986) Automatic construction of diagnostic rules. Fourth Mediterranean Conference on Medical and Biological Engineering, MECOMBE '86, 672-675. Sevilla, Spain.
- [36] Muggleton, S. (1991) Inductive logic programming. *New Generation Computing* 8 (4), 295-318. Ohmsha and Springer-Verlag.

- [37] Muggleton, S.H. and Buntine, W. (1988) Machine invention of first-order predicates by inverting resolution. Fifth Intl. Conference on Machine Learning, 339-352. Ann Arbor, MI: Morgan Kaufmann.
- [38] Muggleton, S.H. and Feng, C. (1990) Efficient induction of logic programs. First Conference on Algorithmic Learning Theory. Tokyo: Ohmsha.
- [39] Neale, I.M. (1988) First generation expert systems: A review of knowledge acquisition methodologies. The Knowledge Engineering Review 3 (2).
- [40] Pearce, D.A. (1988) The induction of fault diagnosis systems from qualitative models. National Conference on Artificial Intelligence, AAAI-88, 353-357. Saint Paul, MN: Morgan Kaufmann.
- [41] Phibbs, B. (1973) The cardiac arrhythmias. St. Louis: The C.V. Mosby Co.
- [42] Quinlan, J.R. (1983) Inferno: A cautious approach to uncertain inference. The Computer Journal 26 (3), 255-269.
- [43] Quinlan, J.R. (1986) Induction of decision trees. Machine Learning 1 (1), 81-106. Kluwer Academic Publishers.
- [44] Quinlan, J.R., Compton, P.J., Horn, K.A. and Lazarus, L (1986) Inductive knowledge acquisition: A case study. Second Australian Conference on Application of Expert Systems. New South Wales Institute of Technology, Sydney, Australia.
- [45] Quinlan, J.R. (1990) Learning logical definitions from relations. Machine Learning 5 (3), 239-266. Kluwer Academic Publishers.
- [46] Shapiro, E.Y. (1983) Algorithmic program debugging. Cambridge, MA: MIT Press.
- [47] Steels, L. (1985) Second generation expert systems. Future Generation Computer Systems 1 (4), 213-221.
- [48] Welbank, M. (1983) A review of knowledge acquisition techniques for expert systems. Ipswich: Martlesham Consultancy Services.