CONTEC^{ES} - AN OBJECT-ORIENTED EXPERT SYSTEM FOR DIAGNOSIS AND REPAIR OF CONCRETE STRUCTURES

CONTEC^{ES} - EIN OBJEKTORIENTIERTES EXPERTENSYSTEM FÜR DIAGNOSE UND INSTANDSETZUNG VON BETONBAUWERKEN

CONTEC^{ES} - UN SYSTEME D'EXPERTS ORIENTE A L'OBJET POUR LE DIAGNOSTIC ET LA REPARATION DE STRUCTURES EN BETON

Gabrielle B. Funk, Hans W. Reinhardt

SUMMARY

Contec^{ES} is an expert system that contains extensive knowledge about concrete technology. It may be used for the diagnosis of deteriorated concrete structures and the selection of appropriate repair measures. It is being developed on the basis of REPCON, which is an expert system prototype of 1991.

Before starting to work on Contec^{ES} aspects of improving REPCON had been collected as far as user friendliness, as well as programming, debugging, and implementation is concerned.

The design of the diagnosing process is now based on KADS which is a systematic methodology for the development of knowledge-based systems.

Contec ^{ES}´ knowledge base is implemented by using Kappa-PC which is an object-oriented and rule-based expert system shell. Kappa-PC´s object-orientation organizes the domain´s terminological data base in a systematic way, and thus helps to reduce the extent of programming.

ZUSAMMENFASSUNG

Contec^{ES} ist ein Expertensystem, das ausführliches betontechnisches Wissen enthält. Es kann bei der Diagnose von schadhaften Betonkonstruktionen und der Erarbeitung von geeigneten Instandsetzungsmöglichkeiten eingesetzt werden. Es wurde auf der Basis des Expertensystems-Prototypen REPCON entwickelt.

Vor der Bearbeitung von Contec^{ES} wurden Verbesserungsmöglichkeiten für REPCON gesichtet, vor allem hinsichtlich der Benutzerfreundlichkeit, der Programmierung, der Fehlerentfernung und des Implementierens.

Der Entwurf des Diagnoseprozesses beruht jetzt auf KADS, das eine systematische Methode für die Entwicklung von wissenbasiertem Systemen ist.

Die Wissenbasis von Contec^{ES} ist mit Hilfe von Kappa-PC implementiert, das eine objektorientierte und regelbasierte Expertensystem-Stelle darstellt. Die Objektorientierung von Kappa-PC organisiert die Begriffsdatenbasis auf der Benutzerebene sehr systematisch, wodurch der Programmierungsaufwand reduziert wird.

RÉSUMÉ

Contec^{ES} est un système d'experts comprenant des connaissances approfondies sur la technologie du béton. Il peut être utilisé pour diagnostiquer des structures de béton détériorées et déterminer des possibilités de réparation appropriées. Il a été développé sur la base du prototype de système d'experts REPCON.

Avant de commencer à travailler sur Contec^{ES} des possibilités d'amélioration de REPCON ont été examinées surtout en vue de la facilité d'utilisation, de la programmation, de l'élimination de défauts et de láccomplissement.

Le projet du procédé de diagnostic est maintenant basé sur KADS qui est une méthode systématique pour le développement de systèmes basés sur la connaissance.

La base de connaissance de Contec ^{ES} est accomplie par le Kappa-PC qui constitue un système d'experts orienté à l'objet et basé sur la règle. Par l'orientation à l'objet du Kappa-PC la base des dates terminologiques est organisée systématiquement de manière à ce que l'ensemble à programmer est réduit.

1. INTRODUCTION

An extensive knowledge about material technological in addition to structural design aspects of reinforced concrete is needed to understand damage to concrete structures. Furthermore, the design of appropriate repair measures requires to follow national regulations which have recently been published in Germany and are not yet very well known to engineers who have to apply them.

Contec^{ES} is based on the idea of REPCON which is an expert system prototype which was developed at Darmstadt University [11]. This project has shown that expert systems are powerful tools to process the enormous amount of knowledge necessary to diagnose concrete deteriorating processes.

REPCON itself could not be developed further as a hardware incompatibility occured. The new start has been used to improve some aspects of this expert system. Difficulties with REPCON like an increasing problem to keep an overview of the programme´s content and as a result of that fact, the problem of debugging the system were caused by the design in a rapid prototyping process. Thus, Contec^{ES} has been designed by applying the KADS methodology which describes an abstract and systematic approach to avoid implementing too complex a design and to ensure that the domain expertise is implemented authentically which means to avoid concessions to specific software abilities. This systematic design of the system also faciliates debugging.

The development tool used for REPCON has not been updated since 1988 which implied that it was a rule-based but not object-oriented shell. The object-orientation of the shell used for the development of Contec^{ES} was necessary to implement the design developed in the abstract and implementation-independent way of KADS.

2. DESIGN CONSIDERATIONS

2.1 Application

The field for application of Contec^{ES} will be the same as for the prior expert system. Once completed - at present it is in the development phase - it willbe a consultant in diagnosing damage to concrete, and rebars in concrete; it will support the selection of appropriate repair measures, and as far as information about the progress of deteriorating processes is available it will predict the further development of the present damage.

2.2 User

The expert system Contec^{ES} helps structural engineers and graduate students as a consultant. Users on this educational level also know the required terminology which is necessary because the used expressions are not described precisely enough by Contec^{ES} to compensate for lacks in the user´s educational background.

Engineers may have a respectable knowledge but the use of Contec^{ES} accelerates the process of diagnosis or the selection of treatment of the defect structure. Contec^{ES} is also a training facility for civil engineers who deal with concrete

repairs less frequently or are not too familiar with current regulations of repair procedures.
Besides the originally intended use of Contec^{ES} as a consultant forengineering tasks it is also used as an education tool for graduate students at Stuttgart University. The students apply and improve their knowledge about concrete damage while working with the programme [4].

2.3 User´s Confidence

To enhance the user´s confidence in the quality of the programme the working-manner of human experts have to be simulated more precisely.

The first impression is very important for the overall judgement of the programme. Right from the beginning the consultation can be tedious because of too many record keeping questions. Thus, the programme-developer has to cut down the number of those very first questions to a minimum [1].

The human expert proceeds by initially asking some few very important questions to get an idea of the problem to be solved. The user gets suspicious about an expert´s skills if too many less specific questions are asked which then in many cases turn out to be irrelevant. Thus, a minimum of preliminary questions should be posed. The selection of these questions is an important and demanding task for the expert system-developer who needs to have an overall understanding of the topic. It is necessary to consider this aspect in the design of an expert system to gain the user´s interest for the programme right from the beginning of a consultation.

Once the user has gained initial confidence, the programme still will be tested - expert-system-users are suspicious as they also should be!
Firstly, the most likely conclusions of the problem should be investigated. Of course, the initially suggested solutions have to differ considerably concerning their likeliness of being true. Otherwise an ordering would not make sense nor would it be possible.

Secondly, a set of questions asked in a row should have a certain connectivity concerning their contents. For that reason, sets of questions have to be formulated which deal with the same hypothesis to be verified. The questioning, then, has to take place by these sets. In order to let the user know when the next hypothesis is to be investigated a short message informs the user about having completed the investigation of one hypothesis and displays the result. Then the user knows that a set of questions will follow which is of a different content.
Thirdly, the programme should not come up with conclusions part way through the consultation. This aspect is especially essential for gaining the confidence of users with a wider knowledge about the topic as they know well themselves what information is necessary to come to a conclusion.

Although the user has to be knowledgeable, still some help should be available in answering difficult questions. The programme eg. should inform the user about special consequences of anwers with considerable significance

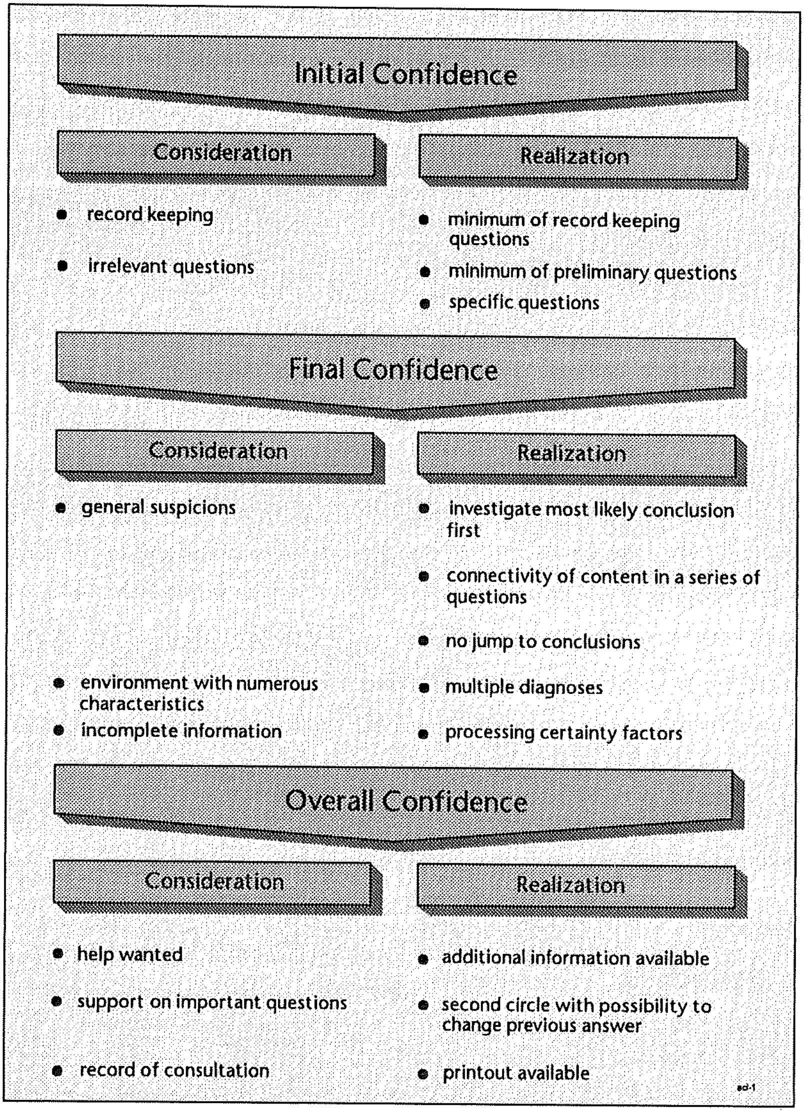

Fig. 1 Gaining the User´s Confidence

which affect the programme flow distinctly. Or some additional information should be mentioned which might be unknown to less experienced users. For example if the programme asks about the presence of chlorides in the structure´s environment the user might not know all possible origins of chlorides besides the ordinary deicing salts and seawater.

In some cases the user might not know that he needs further information on a posed question and does not make use of the help function. Then possibly the programme would have to work with incorrect answers which lead to wrong conclusions, according to the fact that an expert system´s conclusions can not be better than the quality of data provided by the user. Thus, in critical cases in which a wrong answer will mislead the entire rest of the consultation, the user is asked in a second loop, whether he is completely convinced of the correctness of what he answered. In that second loop the user then has the possibility to change the previous answer to lead the programme in the correct direction. These second loops are most important to be added to the preliminary questions which set the course of the programme flow.

These additional features of the programme will lead to the final acceptance of the expert system, and convince the user of the special abilities an expert system ought to have.

2.4 Certainty

Concrete structures are usually exposed to an environment with numerous characteristics and also deteriorations can have several different origins. Concerning this fact we mostly deal with multiple diagnoses in each case. At least all initial hypotheses have to be investigated. In many cases it is a difficult and expensive duty to acquire all necessary data to verify hypotheses. Thus, mostly the programme has to conclude on incomplete information which reduces the certainty of the achieved solutions. Therefore, it is important to also inform the user about the less certain conclusions.

The programme processes certainty factors. Certainty factors (CF) indicate the extent of belief that the conclusion is true by the knowledge of a certain evidence. The range of CF is between -100 and +100. A positive certainty factor is given if the hypothesis is confirmed by a given information. Negative certainty factors indicate that a hypothesis is disconfirmed. Certainty factors are calculated by approaching certainty asymptotically as confirming or disconfirming data is acquired step-by-step [3]. The same as in one of the most popular expert systems called MYCIN [10], which is a medical expert system, also Contec^{ES} combines a degree of belief with the conclusion drawn from a certain evidence, and accumulates CFs during a consultation [5]. Certainty factors were not part of Kappa-PC. Thus CFs had been programmed following the equation used in the expert system shell of REPCON which is called Personal Consultant Plus, developed by Texas Instruments [12].

2.5 Dynamic Data Exchange

To let the user know what data has been the basis of Contec^{ES}' conclusions, the programme has the ability to produce a record of the consultation.

As Kappa-PC is a "Windows"-software it also supports "Windows"-functions like the dynamic exchange of data. Exchanging data dynamically means that one application can send and receive information to and from other applications. An application can execute a command in other applications [8]. In order to produce these printouts a connection of Contec^{ES} via DDE (dynamic data exchange) to another application was developed which is called "Toolbook". Toolbook receives data collected by Contec^{ES} during a consultation and does a formatting before the record is sent to the printer. [1].

3. DEVELOPMENT

3.1 KADS

Among the very few publications on methodologies for the systematic design of knowledge-based systems the best known is the report on two projects of the European Strategic Programme in Research in Information Technology (ESPRIT). Hickman et al. introduce in [7] the KADS methodology which was developed in the two projects "A methodology for the design of knowledge-based systems" (July 1983) and "A methodology for the development of knowledge-based systems" (March 1988).

The KADS methodology defines all phases of development, describes activities which have to be carried out in these phases and the output being produced. The phases known from the conventional software life-cycle are analysis, design, implementation, installation, use, maintenance and knowledge refinement. A model of expertise suggests four distinct types of knowledge that support expertise. These types described in this "Four-Layer Model of Expertise" are represented as the Domain-Layer, the Inference-Layer, the Task-Layer and the Strategy-Layer. This part of KADS concerning the analysis and design phase has been considered in developing Contec^{ES} [6].

In an object-oriented expert system the domain layer can be understood as the object-hierarchy. Static concepts are described which are called "objects". These objects are related to one another hierarchically in an object tree [Fig. 2]. The branches show relations of inheritance which start in the root and end in leaves which are the most specific sort of objects, called "instances".

The "Inference-Layer" describes the inference process in an abstract way by relating "meta-classes" and "knowledge-sources". "Meta-classes" are the roles that the classes, as defined in the "Domain-Layer" play in the process of finding a solution. Examples for "Meta-Classes" are observables like measurable variables, hypotheses like assumed solutions or solutions like diagnoses.
"Knowledge-sources" are procedures which change "input-meta-classes" into "output-meta-classes".

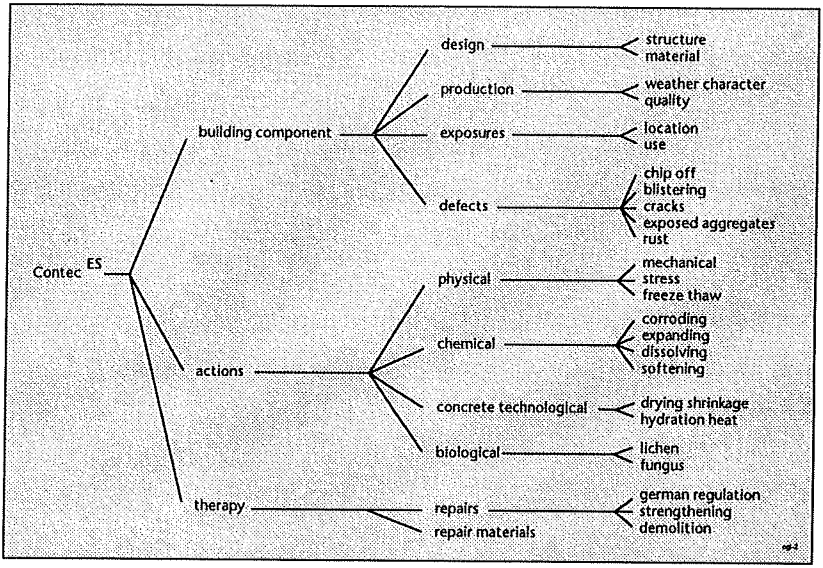

Fig. 2 Object-Tree

While in the inference structure procedures as parts of the whole programme are only described by name, the "Task-Layer" provides the procedural interpretation of these names. In rule-based systems, there are basically two sorts of reasoning processes: "forward chaining", which is event-driven, and "backward chaining" which is solution-driven and the common procedure in diagnosis.

The "Strategy-Layer" specifies circumstances for the sensible application of the task layer. Plans are designed for the way out of deadlocks and for the improvement of the programme flow. Thus, this last step in designing a knowledge-based system represents the strategic reasoning in the problem solving process.

The definite hierarchy of the layers as part of this "Four-Layer Model of Expertise" is achieved by following a certain chronology for the different steps during the design of a knowledge-based system. One layer has to be worked out completely before the elaboration of the following layer can begin. This "Four-Layer Model of Expertise" is then progressing towards the implementable form of expertise, which means that it has to be refined in all its parts before the implementation can begin. The software development environment or programming language must not be chosen before finishing the working-out of the "Four-Layer Model of Expertise" because some types of software do not support the congruent implementation of the design described above. This procedure guarantees the design of the domain characteristics, and therefore improved on the rapid prototyping of REPCON.

For the implementation of Contec^{ES}, an object-oriented and rule-based expert system shell was used. In contrast REPCON is exclusively based on rules [11].

3.2 Object-Oriented-Programming

The constituents of an object-oriented design are classes, methods, messages and inheritance. Winblad et al. [13] suggest a grammatical approach for the identification and definition of classes which means to take nouns of the domain in question as potential identifiers of the classes of objects, and verbs as methods describing the behaviour of classes.

Classes are abstract patterns for specific objects. They describe characteristics and the behaviour of objects which belong to these classes. Subclasses of classes can be defined which inherit all characteristics and methods from their parent class. While classes can have further characteristics than inherited from the parent class, instances are the most specific sort of classes and have inherited characteristics only. Slots are variables which describe characteristics of classes. The behaviour of classes is defined in methods which are procedures. These methods are in most cases activated by accessing slots which means to either initially put in a certain value or to change it. The hierarchy of classes goes along with a hierarchy of inheritance which avoids redundant definitions. Therefore, slots and methods should be defined on the most abstract level which reduces programming work [9].

3.3 Rule-Base

Domain relations are represented in rules. The statements following "if" in a rule are premises, and the statements following "then" are conclusions. Premises indicate when a rule is to be applied and the conclusion indicate the hypothesis which is confirmed or disconfirmed by the knowledge of the facts mentioned in the premise part [8].

Rules have several advantages for diagnosing consultations.

- General knowledge can be used for the examination of specific problems
- Also rarely occuring constellations can be considered; i.e., about which no sufficient statistical data is available
- Modification is easily done as the rules are not explicitly related to one another [2].

These two features of Kappa-PC: the object-orientation and the rule-processing feature were necessary to implement the programme´s design which had been developed initially by applying the "Four-Layer Model of Expertise" as described above [6].

4. IMPLEMENTATION

The design is oriented towards the user requirements. This means that the analysis of the problem-domain and the design of the programme has to be completed before the question of implementation is considered. By following this idea, concessions to a specific implementation form can be avoided.

Besides the above mentioned effect of basing the design on ideas of KADS, debugging is facilitated as the programme flow is divided into precisely defined steps. These steps stand for specific levels of knowledge about a case of diagnosis. Therefore redundancies and incomplete information can be discovered more easily.

These steps are the same in experts´ proceeding which facilitates the acquisition of knowledge in expert interviews as the interview-partner can easily understand how the programme works. This positive effect is even more obvious the less experienced the interview-partner is, concerning computers.

And finally not to forget about the programme developer´s work of eliciting knowledge. The process of extending the programme is a straight forward task. Starting out with the study of literature, the important expressions have to be implemented. Then the new information has to be devided by its level of importance, which is an comparably easy task for a programme developer who is also an expert in the programme´s domain. Finally, the structured information is implemented by using the expressions implemented beforehand. Thus, the analysis of knowledge for implementation is following exactly the same objectives as eg. the abstraction of knowledge for a teacher´s lecture.

5. PROGRAMME CONTENT

The Domain-Layer described in the "Four-Layer Model of Expertise" basically contains expressions describing the domain as mentioned above. Figure 2 shows the object-tree representing these expressions.
The object hierarchy so far consists of three main classes. Subclasses of the main class "Building Component" describe the defect part of the structure in question, its design characteristics, facts that occured during the production, the exposure conditions and the defects. A hierarchy of "Actions" has been implemented which shows an extensive set of physical, chemical and in the production process initiated actions. The branch of the object tree called "Therapy" holds the information on European and German standards, guidelines, and recommendations for repair procedures.

6. PROGRAMME FLOW

In the first step, the task for the consultation has to be chosen which at the present state can be diagnosis, selection of repair and selection of surface treatment. In the second step, some information is collected to get a first hypothesis of the solution. It is efficient to formulate a hypothesis based on the information which is available right away and does not need great efforts to acquire. Thus, the first part of the consultation uses forward chaining

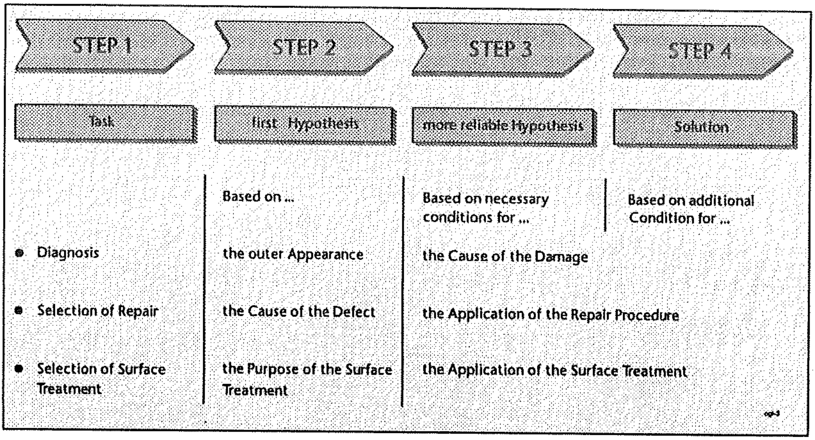

Fig. 3 Programme Flow

inference. Then, necessary conditions for this hypothesis have to be checked to find out whether it deserves more trust. If this is not the case another hypothesis is considered and the process starts all over again. In the last step, additional conditions are investigated to select the solution which is best (eg. least cost-intensive) or most reliable.

During the process of checking further conditions the user is asked for the data which is needed to check specific rules that are assumed to apply according to the information already known. This process is realized by using backward chaining inference.

7. CONCLUSION

In an extended analysis of what Contec^{ES} ought to achieve, characteristics of the work of human experts had been named. Expert systems which are not developed in a rapid prototyping process need considerable efforts of preparing the design. The design should be elaborated step by step. This means that each level of abstraction of the domain has been elaborated separately following the degree of abstraction to not getting lost in problems of specification. Thus, the programme is simulating the work of a human expert precisely which has the following advantanges:

- to enhance the user´s confidence in the programme
- to supply a help-function
- to guide the user through significant questions
- to facilitate debugging

8. RECOMMENDATIONS

The more precise the work before implementation is carried out the better an expert system will be. If the aspects in the analysis, design and development phase described in this paper are considered the quality of a knowledge-based programme and its acceptance by its users can be improved.

9. ACKNOWLEDGEMENT

The support by the German Association for Reinforced Concrete via research grant V 336 is gratefully acknowledged.

10. REFERENCES

1. Allwood R J, Goodier A: Overall Strategies for Diagnostic Expert Systems, Information Technology for Civil and Structural Engineers, Civil-Comp 93, Conference Proceedings, Civil-Comp Press, Edinburgh, UK, 1993, pp 73 - 78

2. Asymetrix Corporation: Toolbook Manual, Bellevue, USA, 1991

3. Buchanan B G, Shortliffe E H: Rule-Based Expert Systems, Addison-Wesley Publishing Company, Reading, Massachusetts, 1984

4. Funk G., Reinhardt H W: Expert Systems for Diagnosis and Repair of Concrete Structures, Computer-Based Training in Property and Construction, Conference Proceedings, University of Salford, UK, 1992, pp 61-67

5. Funk G: REPCON - An Expert System for Diagnosis and Repair of Damages at Concrete Structures, Otto Graf Journal, Vol. 3, 1992, pp 63 - 74

6. Funk G, Reinhardt H W: The Development of an Object-Oriented Expert System for Diagnosis and Repair, Information Technology for Civil and Structural Engineers, Civil-Comp 93, Conference Proceedings, Civil-Comp Press, Edinburgh, UK, 1993, pp 87 - 94

7. Hickman F R, Killin J L, Land L, Mulhall, T, Porter D, Taylor R M: Analysis for Knowledge-Based Systems, Ellis Horwood Limited, Chichester, UK, 1989

8. Intellicorp: Kappa-PC User´s Guide, Intellicorp Incorporated, Mountain View, California, 1992

9. Kurbel K: Entwicklung und Einsatz von Expertensystemen, Springer Verlag, Berlin, 1992

10. Shortliffe E H: Computer-Based Medical Consultations: MYCIN, New York: American Elsevier, 1976

11. Sohni M: Entwicklung eines Expertensystems zur Beurteilung, Beseitigung und Vorbeugung von Oberflächenschäden an Betonbauteilen, Doctoral Thesis, Darmstadt, 1991

12. Texas Instruments: Personal Consultant Plus Reference Guide, Texas Instruments Incorporated, Austin, Texas, 1988

13. Winblad A L, Edwards S D, King D R: Object-Oriented Software, Addison-Wesley Publishing Company, Reading, Massachusetts, 1990