Zeitschrift: IABSE reports = Rapports AIPC = IVBH Berichte

Band: 58 (1989)

Artikel: Knowledge-based systems in civil engineering (from CAD to KAD)

Autor: Hartmann, Dietrich

DOI: https://doi.org/10.5169/seals-44912

Knowledge-Based Systems in Civil Engineering (from CAD to KAD)

Systèmes à bases de connaissance en génie civil

Wissensbasierte Systeme im Bauingenieurwesen

Dietrich HARTMANN
Universitätsprofessor
Ruhr Univ.
Bochum, Fed. Rep. of Germany

Dietrich Hartmann, born in 1944, is Professor for Computational Engineering at the Ruhr University Bochum since 1987. From 1982 to 1987 he was Professor for Structural Mechanics and Optimization at the University of Dortmund.

SUMMARY

Planning and design in civil engineering require an integrated approach based upon a system engineering philosophy. Although numerical methods are very significant for the solution of the problems, engineering expertise and knowledge are much more central. Knowledge based systems (KBS) provide the potential of computerizing the expertise and knowledge of experts in specified knowledge domains. In particular, the problem solving behaviour of engineering experts can be simulated. It is demonstrated by means of distinct CAD/CAE examples what types of mechanisms are required to represent and evaluate engineering knowledge. It turns out that, in the future, a comprehensively hybrid philospohy is needed to obtain knowledge aided design or engineering (KAD/KAE), respectively.

RESUME

La conception et le dimensionnement en génie civil exigent une méthode intégrée basée sur un système d'ingénieur. Les méthodes numériques sont très importantes pour la résolution des problèmes, mais l'expérience et la connaissance sont plus importantes. Les systèmes à base de connaissance donnent la possibilité d'informatiser l'expérience et la connaissance des experts dans des domaines specialisés. On démontre qu'à l'avenir, la philosophie hybride est nécessaire pour obtenir le dimensionnement ou l'ingénierie assistée par les bases de connaissance à l'aide d'exemples concrets.

ZUSAMMENFASSUNG

Planungs-, Entwurfs- und Konstruktionsaufgaben im Bauingenieurwesen, erfordern eine ganzheitliche Sicht im Sinne der Systemtechnik. Numerische Methoden sind zwar eine wichtige Voraussetzung zur Lösung derartiger Probleme; ohne Ingenieursachkompetenz und -wissen aber ist keine ganzheitliche Lösung möglich. Wissensbasierte Systeme (WBS) bieten die Chance, Expertenwissen und damit das Problemlösungsverhalten von Ingenieurspezialisten zu computerisieren. An konkreten Beispielen aus dem Bereich CAE/CAD soll gezeigt werden, dass hybride Wissensrepräsentations- und Verarbeitungsmechanismen erforderlich sind, um das CAE/CAD zukünftig in ein Knowledge-basiertes CAE/CAD (KAE/KAD) überführen zu können.

1. INTRODUCTION

Knowledge based systems (KBS), one of the recent derivates of Artificial Intelligence (AI), provide the potential to incorporate knowledge into those engineering activities that for a long time could not be represented by means of algorithms. In particular, the problem solving behaviour of engineering specialists, being sophisticated in specified knowledge domains, can be captured provided that a sufficiently narrow knowledge domain is considered. Thus, KBS enable us to improve the "brute force computing" to an "inferential computing", an approach that is needed very much in CAD/CAE.

Although, KBS (also called expert systems) are said to be AI-research they are actually not such dramatic as often considered. In fact, KBS are nothing but a new software technology that permits the formalization and representation of knowledge as well as expertise provided that adequate representation mechanisms are available. AI researchers, who are working at the top of AI used to denote KBS as a methodology that no longer can belong to AI because of the tremendous success KBS have had in the past years!

2. REPRESENTATION OF ENGINEERING EXPERTISE

As we all know, engineering expertise plays the central role in engineering much more than numerical methods. While numerical methods are a cornerstone for the accurate analysis of physical or mechanical properties, "engineering-know-how" brings together the parts that really make an integrated system in the sense of the CIM-philosophy discussed all over the world. According to FEIGENBAUM, one of the fathers of AI, expertise and knowledge is characterized by the following: Even though a lot of professional work seems to be expressed in mathematical formulas the matters that set apart experts from beginners are symbolic, inferential and are rooted in experimental knowledge. This makes evident that experts have acquired their expertise not only from explicit knowledge found in textbooks and lectures but also from experience gained by doing things again and again, from eventually learning when to go by the book and when to break existing rules.

If we are able, to formalize and computerize knowledge and expertise, engineering related activities such as

- data analysis and interpretation,
- definition of engineering objects in relation to other objects by means of semantic nets,
- classification, diagnosis, selection, etc.
- formation (planning, modelling, developing, etc.)
- assistance and training,
- evaluation and verification,
- monitoring and control, etc.

can be coupled with traditional computational techniques. Precondition to the integration of numerical and knowledge oriented models is an adequate knowledge representation as already mentioned. Since there is a

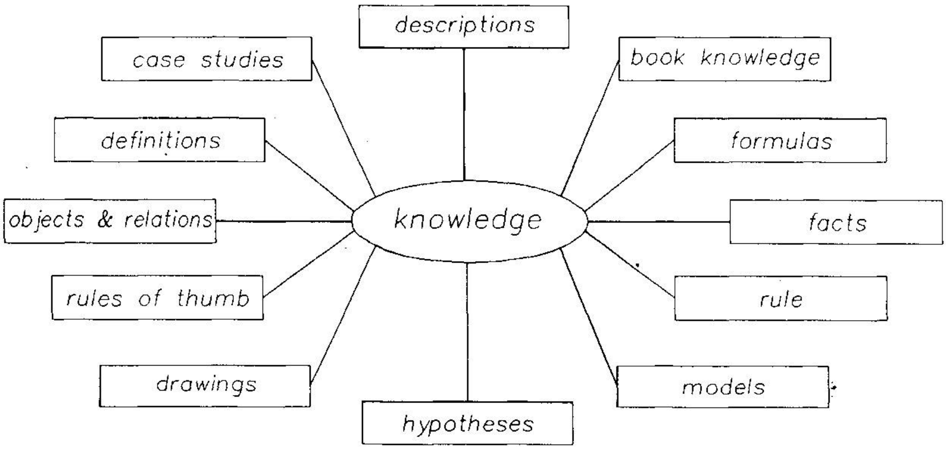


Fig. 1 Knowledge categories

wide variety of distinct knowledge categories (an precise definition is controversial but an enumeration like that in Fig. 1 may be sufficient) it is obvious that also a wide variety of representation mechanisms is mandatory. Present research of KBS indicates that rule based paradigms in association with object-oriented paradigms and blackboard techniques as well as semantic nets (frames, scripts, events, etc.) provide the versatile tool needed for engineering problems. If all of the above mentioned paradigms are combined within one single system this system is called a hybrid system. (Typical examples are KEE, ART, TWAICE, KNOSSOS, just to name a few.)

A further characteristic of KBS is the strict separation between knowledge and operational mechanisms (inference engine) that act on the knowledge and infer new knowledge from existing one. However, apart from these two fundamental components additional components are necessary for practical applications (see. Fig. 2).

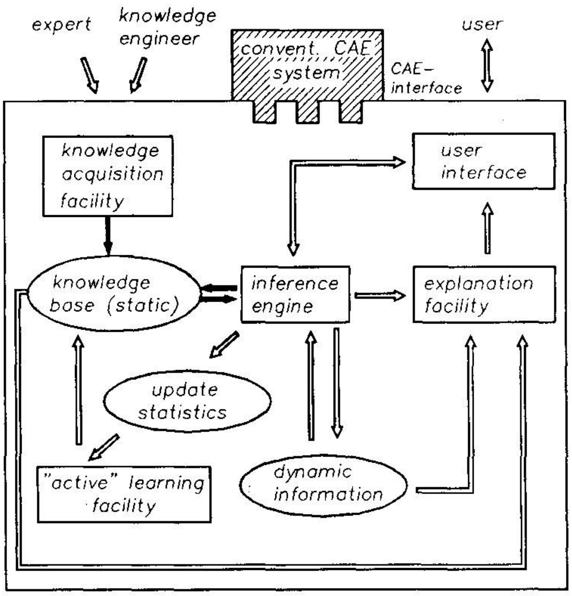


Fig. 2 KBS architecture

Thus, according to Fig. 2 we have a knowledge acquisition facility that supports all activities needed to acquire knowledge in a computer readable format and an explanation facility to make transparent conclusions and inference paths. For real world problems in engineering an interface to existing conventional CAE-software is absolutely necessary. Very recently a learning facility that allows updating of knowledge has appeared. However, active learning adaptation is still a matter of AI research.

Interfacing conventional software with KBS will yield the following scenario in the nearest future (see Fig. 3).

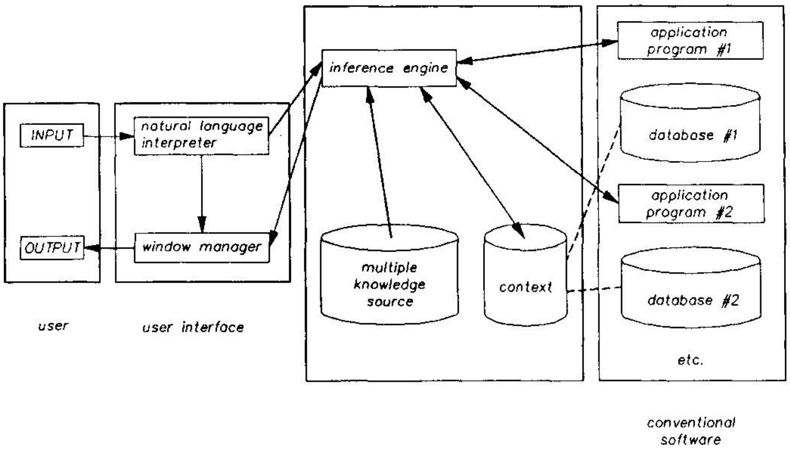


Fig. 3 Conventional CAE coupled with WBS

Fig. 3 indicates that a natural language description may be used for the problem definition. (Natural language systems and pattern recognition are also AI-disciplines that are becoming more and more important to engineering problem solutions.) Fig. 3 also demonstrates that the KBS methodology and conventional software is integrated into one computer system.

3. EMBEDDING ENGINEERING KNOWLEDGE INTO CAD

The computerization of engineering knowledge leads to substantial changes or, at least, major modifications in conventional CAD. The fundamentals of conventional CAD systems, exclusively written in traditional (procedural) computer languages like FORTRAN or C, are graphic oriented entities such as lines, arcs, polygons, cubes, etc. These graphic primitives are well suited to make impressive drawings. However, if design and manufactoring is considered, or the complete horizontal CIM-life cycle of a construction from planning and preliminary design over "final" design to fabrication including management is to take into account, then drawing is only one aspect among others. In this case, a much more sophisticated approach is needed. All over the world, researchers and software specialists in the CAD-domain are aware or getting aware of the significance that the embedding of knowledge into CAD captivates. As a consequence, numerous attempts have been undertaken to incorporate knowledge into CAD. Some of the most interesting categories of realisation will be discussed in the following subchapters.

3.1 INTELLIGENT CAD

When CIM-applications are to be addressed, a CAD system must devise its own methods for defining data "objects" or "entities" and retrieve them from a data base. Processes such as the creation of lists or data structures, however, are not supported by traditional languages. They may be simulated but then traditional programs become very large. Also, such programs are expensive to debug or modify. Therefore, an "object-oriented philosophy" has to be developed to automate the design process efficiently.

From the author's point of view, the ICAD-system [1] is the first CAD-system designed from the ground up to employ an object- or feature-based data structure. The ICAD-system has a language structure that is more suited to the way designs take shape than do traditional languages. As each part of the design is invented, the designer creates the part using standard component features, then defines the rules for connecting the part to the structure or machine. The program which describes the part is called an "object". Thus, objects may be created as they are needed, then linked to other objects in a very natural fashion. The process allows the designer to build his parametric objects part by part, testing each part as it is created. This piecewise approach parallels the design thought process very much.

In contrast to this method, in FORTRAN e.g. the entire program would have to be written, compiled, debugged, linked and loaded before any portion of it could be tested. When traditional languages are applied to automate design work, the engineering of software tends to become more complex than the design itself. This holds particularly for program modifications.

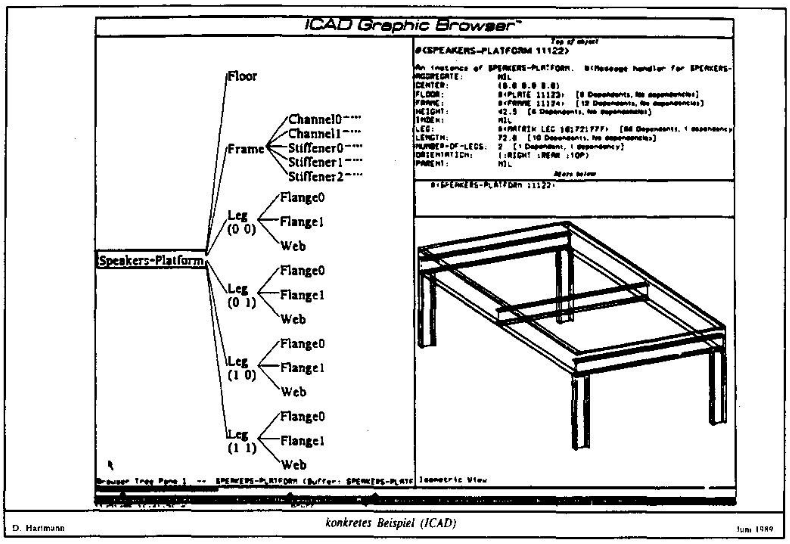
In a general sense, the process of engineering consists to large part in manipulating symbols and linking those symbols into lists in accordance with certain rules. The rules may stem from mathematical formulas or other type of knowledge (e.g. technological requirements such as "bolt X must fit into bolt hole Y"). In order to represent the knowledge there are three basic components in ICAD:

- A symbolic language for product description.
- A graphic browser for viewing and editing the product design.
- A relational query language for retrieving parts from an existing library.

The symbolic language (ICAD) is based upon LISP but easier to handle by engineers, and directly assists the engineering work. The versatility clearly shows evidence that LISP, as an AI-language, fits the demands of an advanced CAD.

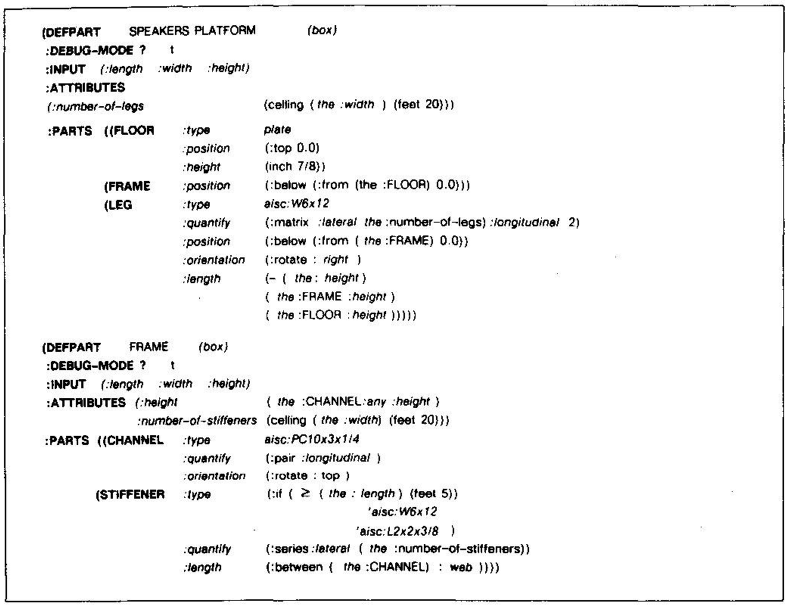
In order to elucidate the ICAD system more precisely, a simple demonstration object (see Fig. 4) with a description of its construction is to be presented. The structure considered represents a simple 3-D-frame structure called "speakers platform".

On the left side the part-whole tree can be seen that depicts the tree hierachy of the construction consisting of a floor, a horizontal frame (grid) and four legs. In the lower right-hand corner one can find the 3-D-representation of the structure as a simple isometric view. At the top right is a window for inspecting the values of the attributes of any object defined.



#<SPEAKERS-PLATFORM 11122>
AGGREGATE: NIL
CENTER: (0.0 0.0 0.0)
HEIGHT: 42.5
INDEX: NIL
LENGTH: 72.0
NUMBER-OF-LEGS: 2
ORIENTATION: (:RIGHT :REAR :TOP)
PARENT: NIL

Fig. 4 Object - oriented ICAD



```
(DEFPART    SPEAKERS PLATFORM          (box)
 :DEBUG-MODE ?    t
 :INPUT  (:length  :width  :height)
 :ATTRIBUTES
 (:number-of-legs                  (celling ( the :width ) (feet 20)))
 :PARTS  ((FLOOR     :type         plate
                     :position     (:top 0.0)
                     :height       (inch 7/8))
          (FRAME     :position     (:below (:from (the :FLOOR) 0.0)))
          (LEG       :type         aisc:W6x12
                     :quantify     (:matrix :lateral the :number-of-legs) :longitudinal 2)
                     :position     (:below (:from ( the :FRAME) 0.0))
                     :orientation  (:rotate : right )
                     :length       (- ( the : height )
                                   ( the :FRAME :height )
                                   ( the :FLOOR :height )))))

(DEFPART    FRAME     (box)
 :DEBUG-MODE ?    t
 :INPUT  (:length  :width  :height)
 :ATTRIBUTES (:height              ( the :CHANNEL:any :height )
             :number-of-stiffeners (celling ( the :width) (feet 20)))
 :PARTS ((CHANNEL    :type         aisc:PC10x3x1/4
                     :quantify     (:pair :longitudinal )
                     :orientation  (:rotate : top )
         (STIFFENER  :type         (:if ( ≥ ( the : length ) (feet 5))
                                               'aisc:W6x12
                                              'aisc:L2x2x3/8  )
                     :quantify     (:series :lateral ( the :number-of-stiffeners))
                     :length       (:between ( the :CHANNEL) : web ))))
```

Fig. 5 Editor window in ICAD

Fig. 5 shows an editor window which contains the design rules for the speakers-platform and demonstrates the way the design rules for the given structure can be established through the ICAD language. Also, ICAD's similarity to LISP becomes evident. In particular, the "DEFPART"-keyword allows the user to define any number of object attributes. Attributes may be orientation, position, length relative to individual parent objects, additional information with respect to fabrication, management etc. The keyword "PART" creates the part-whole hierachy that is graphically represented in the tree display parallel to its creation.

The attributes of any object in the structure can be referred to by any other object through a symbolic reference scheme. Such a reference is stated by means of the word "THE" and contains a path from one object to another. This concept materializes a semantic network of dependencies between objects. Also, this semantic net naturally represents the taxonomy knowledge of structural components within a total construction or building. Furthermore, the application of relations between objects incorporates a form of inheritance between "parent" and "child" objects similar to the inheritance mechanism used in the frame paradigm in expert systems.

Another kind of dependency is created in the part "STIFFENER". Modifications to a reference configuration are set up in terms of production rules of the type:

IF premise THEN conclusion, or
IF event THEN reaction.

In our current example a production rule for the part "STIFFENER" is used to determinate a stiffener:

IF length of stiffener $\geq$ 5 feet
THEN use W6X12 beam
ELSE use L2X2X3/8 angle.

Using more sophisticated conditionals allows arbitrary complexity. Thus, besides object oriented concepts, taxonomies and semantic nets, ICAD also provides the standard rule paradigm of rule based systems. The rule paradigm provides the potential to create general constructional knowledge bases, but company specific knowledge bases are possible as well.

To summarize the features of ICAD, the ICAD-system is based upon knowledge of objects and their reaction to alternations and modifications. Instead of working with absolut data a parametrical design is specified taking into account all possible alternatives. Therefore, the design process can be shaped identical to the natural way a designer proceeds: First, a rough model is designed, then it is refined stepwise to become more and more detailled. The incremental logic of design associated with the possibility to make alternations at any time makes ICAD the most advanced CAD system the author knows.

Despite the advantages of ICAD there are some drawbacks. ICAD does not allow an interactive design procedure exclusively based on graphic modelling. Instead, the object oriented view requires a programming in terms of ICAD commands. In other words, to acquire improved semantics, programming in a LISP-like language is required. The final ICAD design, however, may be transferred to a conventional CAD-system that contains graphic capabilities of a high quality. Another drawback is the high cost of ICAD that currently runs on symbolic-workstations only (total costs about US-$ 250.0000)

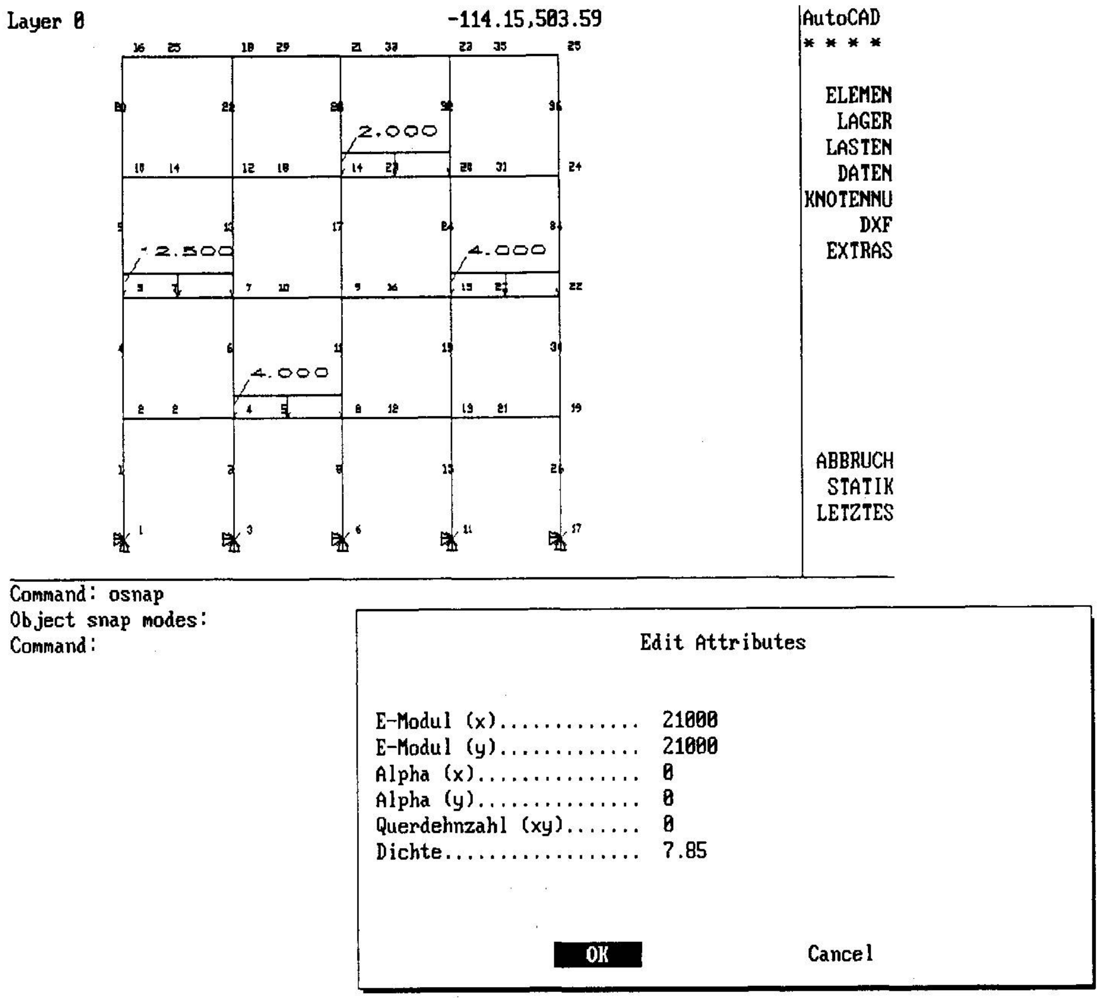
3.2 KNOWLEDGE BASED PROGRAMMING TECHNIQUES

High costs are a crucial obstacle for civil engineers in particular. Therefore, it is clear that in civil engineering all attempts are made to increase the intelligence of existing CAD-systems without producing astronomically high costs. In this context, two examples may exemplify this approach, the first example is the BERT expert system, the second is the research carried out by one of the author's co-workers.

The BERT system [2] is a knowledge based system that links together a conventional CAD design for parts of a brickwork building with a rule based system capturing the standards for brickwork design. Starting from a conventional CAD drawing (AUTOCAD) relevant facts from the drawing or the internal data base are extracted and converted as a context data base for the rule based expert system. The expert system attempts to bring into conformity the given facts with the knowledge incorporated in the knowledge base. The inference engine chains the facts with the rules in a backward chaining manner in order to infer that all the standards hold. If not so, comments and suggestions are given such that an appropriate construction can be created. This cycle continues until the expert system is unable to find anything wrong.

The second example demonstrates that intelligence can also be incorporated in the interior of a conventional CAD system, by enhancing the command and menu structure of a CAD system. In this case the knowledge based character is achieved by AI-language-based programming rather than by creation of a knowledge base or inference engine. Based upon the aforementioned AUTOCAD-system, representing a worldwide quasi standard, a knowledge based pre-processor for structural analysis is in process (see [3]). Starting with an architectural CAD-model a structural system is prepared for structural analysis, where a finite element method is taken as the fundamental computational procedure. It is well known that modelling a finite element model is cumbersome, time consuming and prone to errors, particularily if large scale structures are considered. Therefore, intelligent aids for finite element modelling are desired very much. In this context, intelligence is understood in the sense that CAD-systems link knowledge about structural data and properties with the CAD-geometry. This link is accomplished through AUTOLISP, a derivate from common LISP with an adaptation to AUTOCAD. AUTOLISP is an adequate language for manipulating symbols in terms of lists. The symbols may be words (e.g. AUTOCAD words), numbers or other lists of symbols.This ability makes AUTOLISP very powerfull for augmenting AUTOCAD with intelligent mechanisms.

Just to give a short impression of how AUTOLISP constructs work the structural system in Fig. 6 is considered.



```
Command: osnap
Object snap modes:
Command:
```

E-Modul (x)............ 21000
E-Modul (y)............ 21000
Alpha (x).............. 0
Alpha (y).............. 0
Querdehnzahl (xy)....... 0
Dichte.................. 7.85

Fig. 6 "Intelligent" AUTOCAD

Whereas in conventional CAD only graphic primitives (lines, arcs, strings, etc.) can be identified without any information on operational steps subsequent to the architectural design (like structural analysis, management, fabrication, etc.) AUTOLISP constructs embed corresponding knowledge into the geometrical entities used. The AUTOLISP module for the above example is shown in Fig. 7. It can be seen that through this module the internal geometry is linked with "computational information".

```
.
.
(if (= ELEMTYP "BIEGE2D")
  (progn
    (while (= EMODULX nil)
      (setq EMODULX (getstring "\nE-Modul : "))
    )
    (setq EMODULY EMODULX)
    (while (= ALPHAX nil)
      (setq ALPHAX (getstring "\nAlhpa : "))
    )
    (setq ALPHAY ALPHAX)
    (while (= NUXY nil)
      (setq NUXY (getstring "\nQuerdehnzahl (xy) : "))
    )
    (while (= DICHTE nil)
      (setq DICHTE (getstring "\nDichte : "))
    )
  );progn
);if

(command "layer" "m" (strcat "DAT" (itoa LNR))
  "c" LNR (strcat "DAT" (itoa LNR)) "")
(command "insert" (strcat PFAD "material") "0,0" (/ FAKTOR 10) "" "0"
      ELEMTYP (itoa LNR) EMODULX EMODULY ALPHAX
      ALPHAY NUXY DICHTE)
.
.
```

Fig. 7 AUTOLISP example

The structural data are converted in terms of AUTOCAD block data structures that can easily be evaluated for a subsequent finite element analysis. The evaluation is performed by means of a PROLOG programm (another AI language) to provide a standard format (FEDIS) accepted by a variety of finite element codes. (Of course, the structural knowledge could also be applied in the BERT fashion; a separate expert system could be created to check the appropriateness for finite element input, for instance with respect to input requirements).

4. KNOWLEDGE BASED CONCEPTS IN STRUCTURAL ANALYSIS AND DESIGN

Besides intelligent pre-processors for structural analysis of the type discussed in the previous chapter there are further domains that are well suited for knowledge based systems. However, at first it should be explicitly pointed out that computational methods themselves are not a matter of knowledge based systems. Since computational methods are founded on consistent theories, that are definitely accurate within prescribed application limits, there is absolutely no necessity for knowledge based approaches. That is to say that knowledge based systems are exclusively successful in cases only where

- no accurate theoretical concept is available,
- a diffuse complexity is present or
- a solution has to be based on permanently available expertise acquired in years of training.

Typical categories of problems that qualify are:

- selecting appropriate solution methods,
- monitoring computational processes and
- assisting and consulting in order to navigate complicated phases in computation.

To exemplify the potential that the knowledge based approach presents, again, two application domains are dealt with.

The first application for structural analysis and design is a knowledge based system created for assisting and consulting a mechanical engineer to identify bifurcation or limit points of a given stability problem [4] according the theory of linear elasticity. Stability problems are characterized by the fact that a specified load level may results in various equilibrium conditions depending on the nature of the problem (snap through problem associated with limit points; bifurcation problems yield primary and secondary equilibrium paths, see Fig. 8).

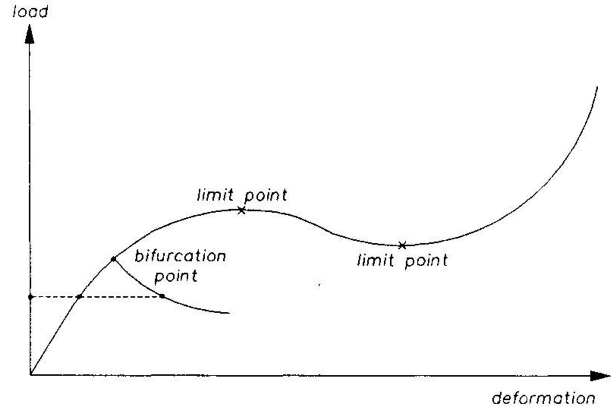


Fig. 8 Stability problems

A knowledge based approach provides the possibility to identify the nature of the critical points (whether they are limit or bifurcation points). The identification process is based upon explicit rules captured in a knowledge base in which new knowledge can be added if detected. In order to draw conclusions during the computational process (e.g. whether a critical point is occuring, what category of point is detected, what method is to be used to procede, etc.) numerical output (e.g. determinant of tangential global stiffness matrix, eigenvalues and eigenvectors) is converted into qualitative facts needed for the inference mechanism applied (modus ponens and backward chaining).

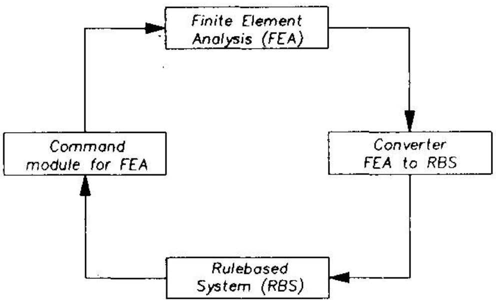


Fig. 9 Architecture of consulting expert

The v. Mises two-bar system illustrates the capability of a knowledge based navigation in complicated computational scenarios:

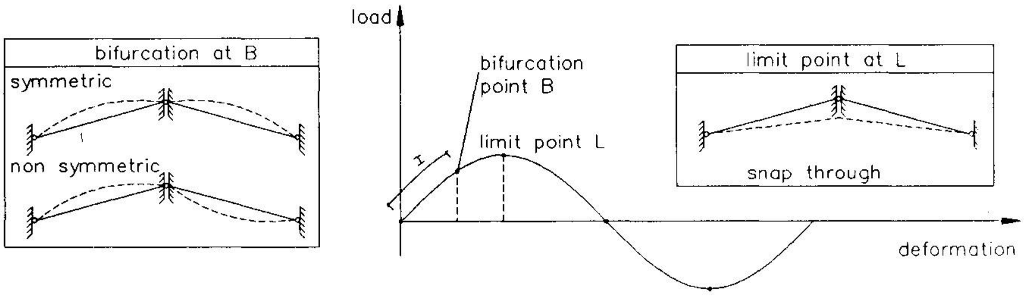


Fig. 10 v. Mises stability problem

The rule based system (RBS) identifies critical points and does consulting in the following fashion (short form) according to Fig. 10:

computational phase	RBS reaction
within part I of the computation:	according to rule # XXX instability has to be expected, confidence 100;
after exceeding point B	according to rule # YYY instability active; confidence 100; explanation: negative diagonal element in the tangent stiffness matrix measures taken: check on category of instability thru eigenvalue evaluation; switch over to arc-length method and back iterate to point B
after a while according to eigenvalue and eigenvectorcheck	instability due to bifurcation
etc.	etc. .....

Although the sample test is elementary it demonstrates that, along with knowledge based systems, computational mechanics is developing from the rather "brute force technique" to a more "inferential computing", as mentiosed at the beginning.

The second application for structural analysis and design addresses the post-processing of structural analysis. Post-processing is not only restricted to the customarily used graphical representation of computational results, in addition, the structural component design is ascribed to post-processing.

The major part of structural component design is knowledge based because the design itself has its roots in standards that contains a diversity of knowledge formats (facts, rules, tables, formulas, comments, figures, etc.). In this context, the research carried out at the civil engineering department of the CMU (Carnegie Mellow University, Pittsburg, PA, USA) under supervision of Prof. Fenves, deserves particular mention. The Standards Processing Expert (SPEX) [5] links numerous distinct knowledge sources within the design process (book knowledge, knowledge on objects, experience, standards) by virtue of a blackboard (see Fig. 11). A blackboard is a central medium with which separate knowledge sources (they may even be written in different languages) communicate. The inference engine schedules the flow of conclusions in the blackboard and monitors the various activities of the knowledge source.

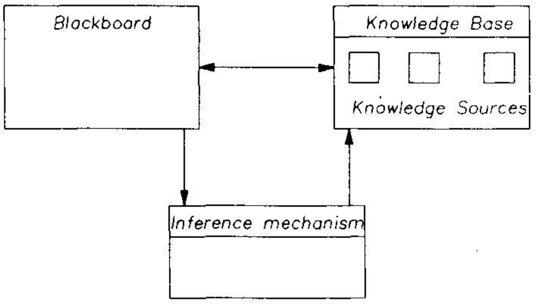


Fig. 11 Blackboard architecture

Current research of one of the co-workers of the present author (see [3]) focuses on the computerization of the new DIN18800, part 2 (stability problems in steel structures). The given standards are transferred to a rule based system utilizing PROLOG-production rules. Currently, also an object oriented expert system shell (TWAICE) is examined. Just to give an insight in the PROLOG rule base a rudimentary PROLOG representation of a production rule (in PROLOG called implication) is given in Fig. 12

DIN 18800 Teil 2 (Entwurf)

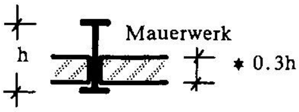


Bild E303.
Aussteifung durch Mauerwerk

3.3.2 Behinderung der Verformung

Behinderung der seitlichen Verschiebung

308 - Ausreichende Behinderung der seitlichen Verschiebung ist vorhanden bei Stäben, die durch ständig am Druckgurt anschließendes Mauerwerk ausgesteift ist, dessen Dicke nicht geringer iat als die 0.3fache Querschnittshöhe des Stabes.

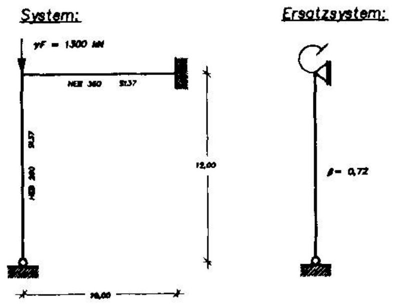
PROLOG

```
% . . . Aussteifung durch Mauerwerk ( 3.3.2.308 )
nachweis (abschnitt_3_3_2,Achse,Nquer) :
,  global (querschnitt,i (_))
,  eingabe_num ('Mauerwerksdicke am Druckgurt','cm',Dicke)
,  global (profilhoehe,H)
,  Dicke >= 0.3 * H
,  Nquer is 0
.
```

Fig. 12 PROLOG production rule example

Based upon production rules of the above type a DIN-adequate check on stability is performed through the backward chaining mechanism incorporated in the PROLOG interpreter (Arity-PROLOG). A PROLOG session in Fig. 13 indicates how the knowledge based system works for a elementary test sample.

In the test sample (the frame system is transferred to a single bar object with a centric compressive load according to the "Einzelstabverfahren" in DIN 18800) the input for the PROLOG post-processor is interactively accomplished (for practical application the input has to be taken from an output file of a finite element run or a corresponding structural analysis programm).



```
?- din18800 t2(Y,Z).
Material [st37,st52,ste460,ste690] :   st37.
Querschnittstyp [kreis,kreisrohr,rechteck,l(geschweisst),kasten(geschweisst),
ipb,heb] :ipb.
Profil [100,120,140,160,180,200,220,240,260,280,300,320,340,360,400,450,
500,550,600,650,700,800,900,1000] :  300.
Knicklaenge um y [m] :        12.
Knicklaenge um z [m] :        12*0.72.
Gamma-fache Schnittkraefte links
    Normalkraft [kN]   :      1300.
    Moment um y [kN*m] :  0.
    Moment um z [kN*m] :  0.
Gamma-fache Schnittkraefte rechts
    Normalkraft [kN]   :      1300.
    Moment um y [kN*m] :  0.
    Moment um z [kN*m] :  0.
**********************************************************
***  Ergebnisse:                       *****
**********************************************************
Abschnitt 3.2.1: nquer(y) -> 0.66506616
           nquer(z) -> 0.95088999
Abschnitt 3.2.2: nquer(y) -> 0
           nquer(z) -> 0
?-
```

Fig. 13 PROLOG session for checking expert system

5. CONCLUDING REMARKS

In the preceding chapters distinct application fields of knowledge based concepts have been tackled. It was intended to demonstrate that knowledge based approaches have many facets and may be materialized in different ways. Despite the diversity of candidate representation formats and practical realisations, conventional CAD/CAE will definitely augmented and modified from "simple" 1st-generation CAD towards more sophisticated 2nd-generation CAD.

REFERENCES:

1. ROSENFELD L.,
   Breaking through the Complexity Barrier Internal ICAD-Paper, ICAD, Incorporated, Cambridge, Massachusetts, USA, 1986
2. BOWEN J.A:, CORNICK T.C., BULL S.P.,
   BERT - An Expert System for Brickwork Design. Internal Paper, Department of Computer Science and Construction Management, UniversitY of Reading, England, 1986
3. HARTMANN D., CASPER E., LEHNER K., SCHNEIDER H.-J.,
   Einsatz von Expertensystemen im Umfeld von Finite-Element-Applikationen. Tagung Finite Elemente in der Baupraxis, RU-Bochum, März 1988
4. TARNOW N.,
   Das Expertensystem GENIUS, Interaktive Steuerung nichtlinearer, struckurmechanischer Lösungsalgorithmen im Finite-Element-Programm PCFEAP Diplomarbeit am Institut für Baumechanik (Prof. Stein), Universität Hannover, 1987
5. GARRET J.H., FENVES S.J.,
   A Knowledge Based Standards Processor for Structural Component Design. Report # R-86-157, Carnegie Mellon University, Dept. of Cicil Engineering, Pittsburgh, Pennsylvania, Sept. 1986