Knowledge-based connection design in steel structures

Autor(en): **Gelder, J.T. De / Steenhuis, C.M. / Huijsing, A.P.**

Objekttyp: **Article**

Zeitschrift: **IABSE reports = Rapports AIPC = IVBH Berichte**

Band (Jahr): **72 (1995)**

PDF erstellt am: **26.04.2024**

Persistenter Link: https://doi.org/10.5169/seals-54651

Nutzungsbedingungen

Die ETH-Bibliothek ist Anbieterin der digitalisierten Zeitschriften. Sie besitzt keine Urheberrechte an den Inhalten der Zeitschriften. Die Rechte liegen in der Regel bei den Herausgebern.
Die auf der Plattform e-periodica veröffentlichten Dokumente stehen für nicht-kommerzielle Zwecke in Lehre und Forschung sowie für die private Nutzung frei zur Verfügung. Einzelne Dateien oder Ausdrucke aus diesem Angebot können zusammen mit diesen Nutzungsbedingungen und den korrekten Herkunftsbezeichnungen weitergegeben werden.
Das Veröffentlichen von Bildern in Print- und Online-Publikationen ist nur mit vorheriger Genehmigung der Rechteinhaber erlaubt. Die systematische Speicherung von Teilen des elektronischen Angebots auf anderen Servern bedarf ebenfalls des schriftlichen Einverständnisses der Rechteinhaber.

Haftungsausschluss

Alle Angaben erfolgen ohne Gewähr für Vollständigkeit oder Richtigkeit. Es wird keine Haftung übernommen für Schäden durch die Verwendung von Informationen aus diesem Online-Angebot oder durch das Fehlen von Informationen. Dies gilt auch für Inhalte Dritter, die über dieses Angebot zugänglich sind.

Ein Dienst der *ETH-Bibliothek*
ETH Zürich, Rämistrasse 101, 8092 Zürich, Schweiz, www.library.ethz.ch

http://www.e-periodica.ch

Knowledge-Based Connection Design in Steel Structures

Projet et calcul, basé sur la connaissance, d'assemblages en construction métallique

Wissensbasiertes Entwerfen von Anschlüssen im Stahlbau

J.T. DE GELDER
TNO Building & Constr. Res.
Delft, The Netherlands

C.M. STEENHUIS
TNO Building & Constr. Res.
Delft, The Netherlands

A.P. HUIJSING
TNO Building & Constr. Res.
Delft, The Netherlands

SUMMARY

A complex task in the design process of steel structures is the design of the connections between the members. To support the structural engineer in the connection design process, software has been developed that is able to calculate the capacity of the connection based on its layout. This software, however, does not support the design of the layout itself. The structural engineer has to enter and modify the layout using his own expert knowledge and experience. Due to lack of knowledge, this often leads to connections which are more expensive to manufacture than necessary. This paper describes an approach to develop a knowledge-based system that supports a structural engineer in designing more efficient and cheaper connections in steel structures.

RÉSUMÉ

Le projet et le calcul d'assemblages des cadres métalliques est une activité complexe. Afin d'assister le projeteur, des programmes ont été développés, lesquels permettent de calculer la résistance d'un assemblage sur la base de sa forme. L'ingénieur doit cependant décider de la forme sur la base de ses propres connaissances et expériences. Un manque de connaissances conduit souvent à des assemblages trop coûteux. L'article décrit le développement d'un système de connaissances permettant à l'ingénieur de concevoir et réaliser des assemblages de cadres métalliques plus efficaces et moins chers.

ZUSAMMENFASSUNG

Der Entwurf von Anschlüssen im Stahlrahmenbau ist eine komplexe Tätigkeit. Um den Konstruktionsingenieur bei dieser Aufgabe zu unterstützen, wurden Programme entwickelt, die aufgrund des Layouts die Tragfähigkeit eines Anschlusses berechnen können. Der Ingenieur muss jedoch immer noch aus Fachwissen und Erfahrung heraus das Layout eingeben und gegebenenfalls abändern. Mangelndes Wissen führt oft zu Anschlüssen, die unnötig teuer in der Fertigung sind. Der Beitrag beschreibt die Entwicklung eines wissensbasierten Systems, das dem Ingenieur hilft, effizientere und daher billigere Stahlbauanschlüsse zu entwerfen.

1. INTRODUCTION

Connection design is a knowledge intensive task in the design process of steel structures. Knowledge is required about mechanical and economical aspects of connections. The connection design process is usually performed by the steel fabricator, who's knowledge about connection design is limited. This paper describes how knowledge technology can help to complete this limited knowledge. To explain the specific problems concerning connection design the process of designing steel structures and connections in particular is explained in chapter 2. Chapter 3 describes how conceptual modelling of knowledge is performed by the Centre of Knowledge Based Systems at TNO Building and Construction Research (TNO-KBS). Chapter 4 describes a prototype knowledge based system for connection design developed at TNO-KBS. Finally, chapter 5 presents some concluding remarks.

2. CONNECTION DESIGN IN STEEL STRUCTURES

2.1 Design process of steel structures

In general, the design process of a steel frame consists of 7 steps (see figure 1). In the modelling phase (step 1), the engineer models the joints as pinned or rigid. Pinned joints are capable of transmitting the forces calculated in design, without developing significant moments which might adversely affect the beams or columns in the frame. A rigid joint has no influence on the distribution of internal forces and moments in the frame, nor on its overall deformation.

1. Mechanical modelling of the frame in the building.
2. Estimation of loads.
3. Pre-design of beams and columns.
4. Determination of forces and displacements in the frame.
5. Check of beams and columns in limit state conditions.
6. If required, adjustment of beams and columns (continue with step 4).
7. Design of joints.

Fig. 1 Design process of a steel frame

Normally global frame analysis is carried out with first order elastic analysis [1]. Input for this analysis are the loads and the stiffness of beams and columns. Output are the deflexions of the frame and the force distributions. The member sizes and the forces which should be transmitted by the joints are the starting point of the design of joints (step 7). The purpose of this design is to find a layout capable of transmitting the forces between the beam and the column (figure 2). This design process is commonly used in various European countries. It follows a practice in which the engineer designs the members and the steel fabricator the connections.

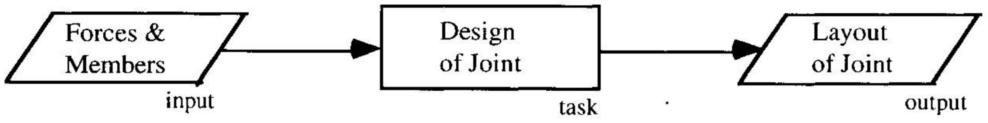
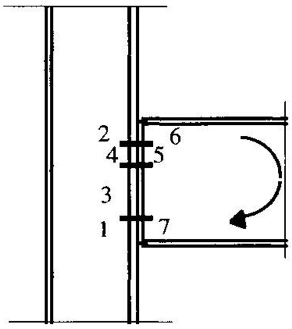


Fig. 2 Design task of steel fabricator

This paper focuses on moment joints and assumes that the rigidity of the joints is anyhow sufficient, and that a joint should be designed for strength only. The here given approach of conceptual modelling of connection design could be extended to rigidity checks. This is, however, not essential for the approach and therefore omitted in this paper.

2.2 Connection design with Eurocode 3 and CASTA/Connections

Eurocode 3 [2] gives rules for the determination of strength, stiffness and rotation capacity of beam-to-column joints. These rules are based on a so-called component-approach. A joint is divided into components. Figure 3 shows the components for a beam-to-column end plate joint.

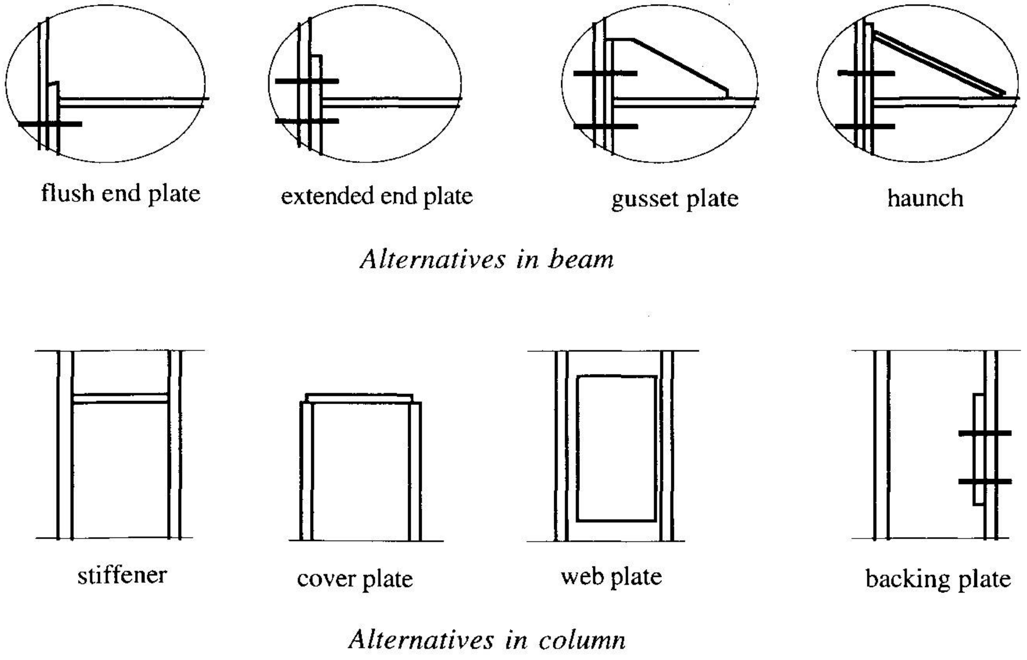


- column web in compression (1), tension (2) and shear (3);
- column flange in bending (4);
- end plate in bending (5);
- beam web in tension (6);
- beam web and flange in compression (7).

Fig. 3 Components in a beam-to-column end plate joint.

The mechanical properties of the components can be determined with the rules given in Eurocode 3. The properties of the joint should be determined based on the properties of the components. Eurocode 3 provides freedom to choose whatever layout of a joint is the best. The drawback of this freedom is that many potential failure modes in a joint need to be checked. The determination of connection properties is consequently a comprehensive task. Programs have therefore been developed for the determination of joint properties according to Eurocode 3 [3].

One of the programs developed is CASTA/Connections [4]. Joint-types covered are: bolted end-plate, cleated and base plate connections between I and H shaped sections. In this paper we focus on its potential for end plate beam-to-column joints. Figure 4 shows some possible alternatives CASTA/Connections can deal with.



Alternatives in beam

Alternatives in column

Fig. 4 Some alternatives CASTA/Connections can deal with

CASTA/Connections calculates a moment-rotation curve of an end plated joint, or checks if a joint is capable to transmit the applied forces. It can be used as follows (figure 5): (1) A designer inputs the layout of a joint into the program. (2) After acceptance of the layout, the program calculates the mechanical properties. (3) The program reports the mechanical properties and the first and second component that will collapse.

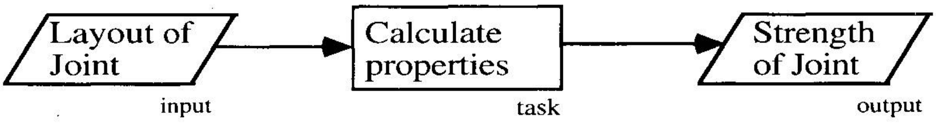


Fig. 5 Design task supported by CASTA/Connections

The task supported by CASTA/Connections is only a subtask of the design task of the steel fabricator. In figure 6 the design task of the steel fabricator (see figure 2) has been decomposed into its subtasks. It shows that the design task is an iterative process. In this process CASTA/Connections supports the subtask *Calculate Properties*. The steel fabricator has to perform the subtasks *Initialize Layout Joint* and *Modify Layout*. This task requires knowledge of mechanical and manufacturing aspects, and is hard to automate because of its creative nature and large number of alternative solutions. A risk is, however, that sub-optimal solutions are achieved due to the limited design knowledge of a practitioner. A solution to this problem is a knowledge based system that can be used by a practitioner to complete his knowledge and that helps to support the design of cheaper connections. In chapter 4 a prototype of such a knowledge based system is described.

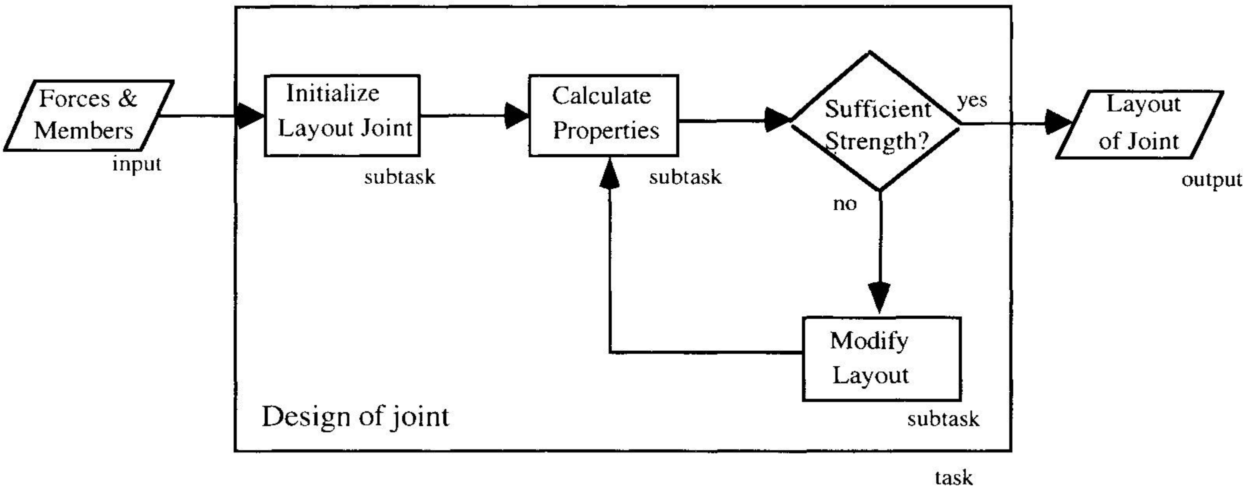


Fig. 6 Design task of a steel fabricator divided into subtasks

3. CONCEPTUAL MODELLING OF KNOWLEDGE

The application of Knowledge Based Systems in practice is growing and will continue growing into the next decades [5]. In the development of knowledge based systems conceptual modelling of knowledge is the main activity [6]. It is of vital importance for the success of the knowledge based system that the knowledge in the system reflects reality and is modelled correctly, completely and consistently. The development of conceptual models requires:

(1) a conceptual modelling theory
(2) a conceptual modelling language that is compatible with the conceptual modelling theory
(3) a conceptual modelling tool that supports the development of conceptual models using the conceptual modelling language

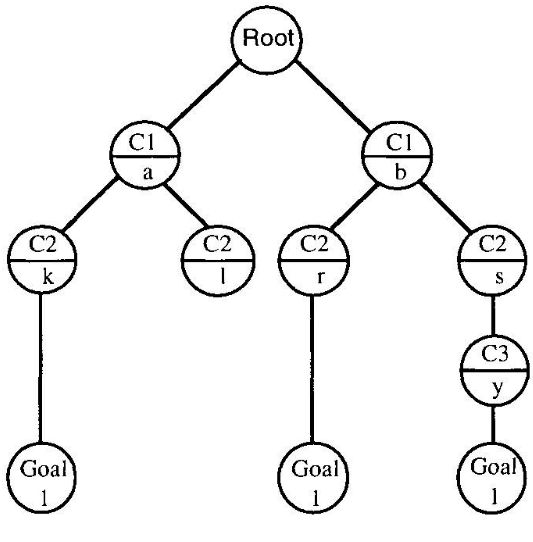
3.1 Conceptual modelling theory: Theory of functional classifications

At TNO-KBS knowledge is modelled following a theory, called the theory of functional classifications [6,7,8,9]. In this theory conceptual modelling of knowledge is viewed as a process in which concepts are modelled following a goal-oriented approach. Only knowledge which is necessary for the goal should be part of the conceptual model.

Often there are several alternatives to reach a specific goal or to describe a concept. In the theory of functional classifications this is called functional equivalence. In a conceptual modelling process a knowledge engineer should try to find functional equivalent solutions in order to get a complete conceptual model. Functional equivalence can appear in three different ways. This is illustrated by figure 7.

3.2 Conceptual modelling language: Decision Tables and Prolog

In the development of knowledge based systems often the language of production rules is used to represent knowledge. This is mainly because most of the commercially available knowledge based system shells are using production rules. Production rules are very powerful to represent knowledge. However, production rules have the drawback that their correctness, completeness and consistency is hard to validate, especially in case of large knowledge bases containing hundreds or thousands of rules. Another drawback is that knowledge bases of production rules are hard to maintain.



1. Under certain conditions additional descriptors may become important to reach the goal. If C1=b and C2=s then C3 becomes a descriptor. If C1=a C3 is not relevant at all.
2. Categorisations of descriptors influence each other. If C1=a then the classification of C2 is (k,l). If C1=b then the classification of C2 is (r,s). This phenomenon is called conceptual interaction between descriptors.
3. Within a goal-constructed category descriptors may have different values. If the values k1 and k2 fall within the category k then a value k1 for C2 is equivalent to a value k2 for C2 if C1=a.

Fig. 7 Appearances of functional equivalence

At TNO-KBS a combination of Decision Tables (DT's) and Prolog is used as a Conceptual Modelling language [9]. Figure 8 shows an example of a DT. A DT is divided into four components separated by double lines. The component left of the vertical double line is called the stub. Above the horizontal double line the stub contains condition parameters (descriptors) and below the horizontal double line action parameters (goals or concepts). The components right of the vertical double line contain the condition categories (above the horizontal double line) and action categories (below the horizontal double line).

In contrast with production rules correctness, completeness and consistency of knowledge represented in DT's can easily be validated because of the structured representation of knowledge. Another benefit is that the expert in the knowledge domain can easily read the model and validate its correctness and completeness [10]. Further DT's have the same power as production rules, because DT's actually are structured production rules. The DT in figure 8, for instance, represents 5 production rules.

Goal 1					
C1	a		b		
C2	k	l	r	s	
C3	-	-	-	not y	y
Goal 1	X	-	X	-	X
	R1	R2	R3	R4	R5

(1) IF (C1 = in category a) AND (C2 = in category k) THEN (Goal 1 is reached)
(2) IF (C1 = in category a) AND (C2 = in category l) THEN (Goal 1 is not reached)
(3) IF (C1 = in category b) AND (C2 = in category r) THEN (Goal 1 is reached)
(4) IF (C1 = in category b) AND (C2 = in category s) AND (C3 = not in category y) THEN (Goal 1 is not reached)
(5) IF (C1 = in category b) AND (C2 = in category s) AND (C3 = in category y) THEN (Goal 1 is reached)

Fig. 8 A Decision Table as a set of structured production rules

The Decision Table language is compatible with the theory of functional classifications. The use of DT's commands a goal oriented approach. The goal is normally represented as an action parameter in a DT. Condition parameters and categories have to be found which are necessary to reach the goal. Functional equivalence can easily be represented in DT's. The DT in figure 8 contains all three appearances of functional equivalence. Rule 4 and rule 5 illustrate that C3 becomes important as an additional descriptor under certain conditions. Conceptual interaction is represented in the influence of C1 on C2's categorisations. And the notion that a value of s1 or s2 for C2 doesn't affect the reaching of the goal illustrates that different values may fall within one goal-constructed category.

Knowledge is usually represented in an hierarchy of DT's. A condition parameter in one table can be an action parameter in another table. In this case the condition parameter is actually a subgoal that has to be reached first in order to reach the ultimate goal. In Artificial Intelligence this process of finding a value for a goal parameter (i.e. the action parameter of the main table) is known as a backward chaining process.

There are some types of knowledge which are not easily represented by DT's [11]. Examples are recursive processes, procedural functions, unconditional decisions and database facilities. For these types of knowledge Prolog is a powerful language. Actually Prolog is a language that is able to represent knowledge in a way compatible to all aspects of the theory of functional classifications, but validation of knowledge represented in Prolog is much more difficult than validation of knowledge in DT's. However, in combination Decision Tables and Prolog is a powerful language to represent knowledge.

3.3 Conceptual modelling tool: Advanced Knowledge Transfer System (AKTS)

At TNO-KBS a knowledge modelling tool has been developed called Advanced Knowledge Transfer System (AKTS) [9]. In AKTS knowledge is modelled in DT's and Prolog. AKTS has graphical editing facilities to build an hierarchy of DT's. Further, Prolog statements can be used anywhere and programs in Prolog or other languages can be called from various places in a DT.

Knowledge included in AKTS can be consulted if one of the parameters is defined as the goal parameter. In a consultation process AKTS tries to find a value for the goal parameter by trying to find values for subgoals, sub-subgoals, etc. The system tries several consequent steps to find a value for a specific parameter:

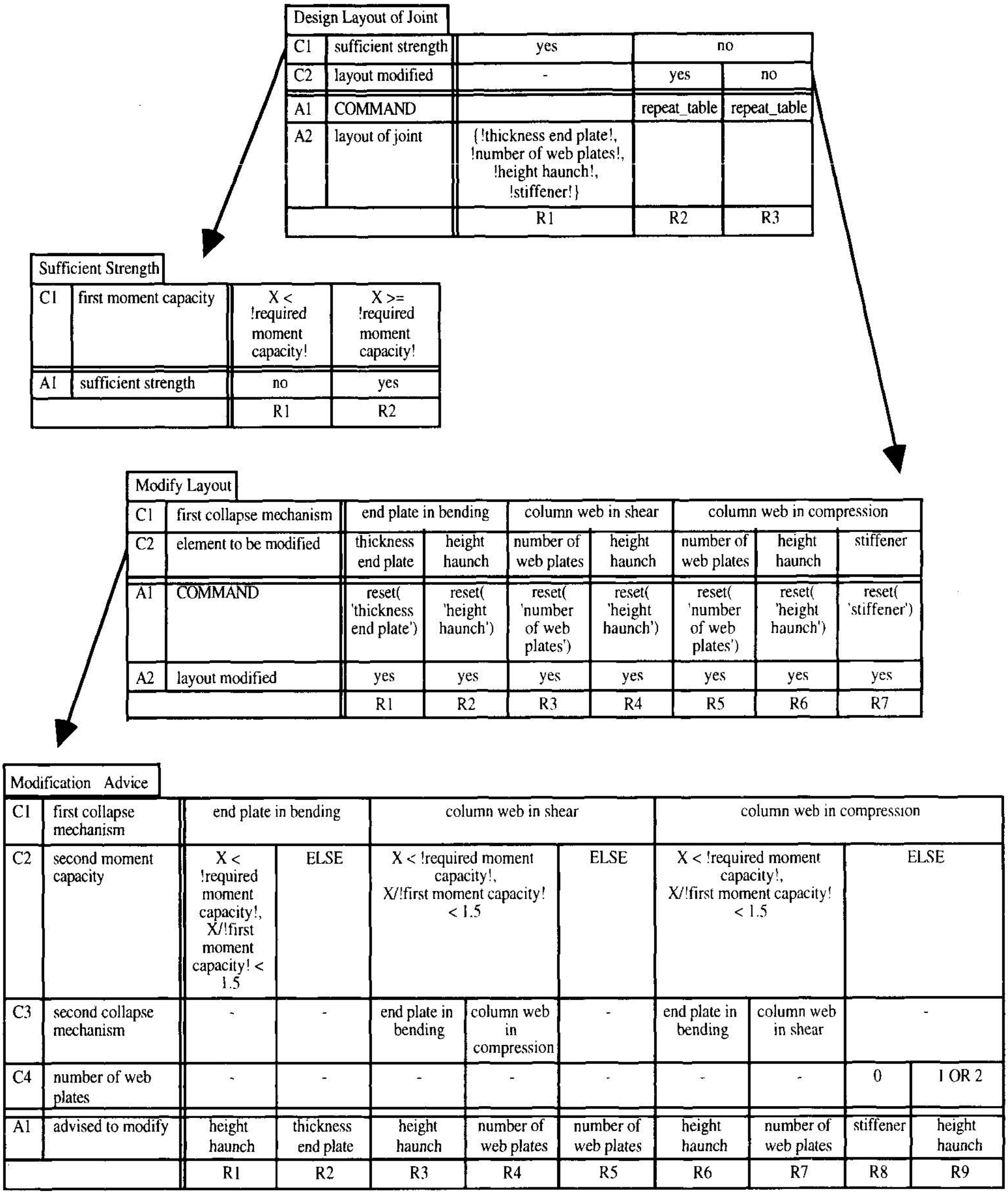
(1) it starts a WHEN NEEDED demon (if defined for the parameter) that might calculate a value for the parameter or read a value from a database. Besides delivering a value for the parameter a WHEN NEEDED demon can also be used to execute a command, for instance, to show a picture or to set initial values of other parameters;
(2) if a WHEN NEEDED demon isn't defined or doesn't deliver a value for the parameter, the system looks for a table having the parameter as action parameter;
(3) if no table is found having the parameter as action parameter, the system reads the default value for the parameter (if defined);
(4) if no default is defined finally the user is asked to provide a value. If the parameter is defined as ASK FIRST the user is asked to provide a value first before the system tries the other strategies.

4. PROTOTYPE KNOWLEDGE BASED SYSTEM FOR CONNECTION DESIGN

The aim of the Knowledge Based System for connection design is to support a designer in performing the subtask *Modify Layout* in the design process of end plate beam-to-column joints (see figure 6). This aim differs from the aim other researchers using Decision Tables for building applications have [10,11,12,13,14]. They use Decision Tables to represent the design standard. Their systems support the design subtask *Calculate properties* and the *Sufficient Strength* check (see figure 6). Although we agree that representing design standards into Decision Tables has great advantages in terms of clearness, flexibility and maintenance [10,12,13], we didn't do this for the practical reason that we already had a computer program available in which the design standard was hard-coded: CASTA-Connections, Therefore we could concentrate on supporting the real creative design task. For a designer this task is complex because the design standard Eurocode 3 provides freedom to choose whatever layout of a joint is the best. Only the requirements defined in the design standard need to be satisfied. This implies that in a specific situation a lot of alternative solutions can be valid. The prototype system supports the designer to find among the alternatives a solution that satisfies the design standard and is optimal in terms of economy. To reduce complexity the knowledge in the prototype is limited to only 3 of the 7 components defined in chapter 2: end plate in bending, column web in shear and column web in compression. Further the number of alternative elements to select is limited as well. Only haunch, stiffener and web plate are included. Figure 9 displays the system of DT's.

4.1 Decision Table *Design Layout of Joint*

The parameter *layout of joint* in DT *Design Layout of Joint* has been assigned the property MAIN GOAL (see figure 10). This means that the goal of the system is to find a value for this parameter. In order to find a value for the goal parameter *layout of joint* the system first executes the WHEN NEEDED demon defined for this parameter (see description of consequent steps to find a value for a parameter in 3.3). The demon defines an initial simple layout: a thin endplate to connect the column and the beam and no other elements. It represents the subtask *Initialize Layout Joint* in the design process (see figure 6). Since no value for the goal parameter has been traced yet the system looks for a DT having the goal parameter as action parameter. The system finds DT *Design Layout of Joint* and starts tracing the first condition parameter *sufficient strength.* This parameter becomes a subgoal. The parameter *sufficient strength* doesn't have a WHEN NEEDED clause, so the system looks for a DT having *sufficient strength* as action parameter. It finds DT *Sufficient Strength.*



Design Layout of Joint

C1	sufficient strength	yes	no	
C2	layout modified	-	yes	no
A1	COMMAND		repeat_table	repeat_table
A2	layout of joint	{!thickness end plate!, !number of web plates!, !height haunch!, !stiffener!}		
		R1	R2	R3

Sufficient Strength

C1	first moment capacity	X < !required moment capacity!	X >= !required moment capacity!
A1	sufficient strength	no	yes
		R1	R2

Modify Layout

C1	first collapse mechanism	end plate in bending		column web in shear		column web in compression		
C2	element to be modified	thickness end plate	height haunch	number of web plates	height haunch	number of web plates	height haunch	stiffener
A1	COMMAND	reset('thickness end plate')	reset('height haunch')	reset('number of web plates')	reset('height haunch')	reset('number of web plates')	reset('height haunch')	reset('stiffener')
A2	layout modified	yes	yes	yes	yes	yes	yes	yes
		R1	R2	R3	R4	R5	R6	R7

Modification Advice

C1	first collapse mechanism	end plate in bending		column web in shear			column web in compression			
C2	second moment capacity	X < !required moment capacity!, X/!first moment capacity! < 1.5	ELSE	X < !required moment capacity!, X/!first moment capacity! < 1.5		ELSE	X < !required moment capacity!, X/!first moment capacity! < 1.5		ELSE	
C3	second collapse mechanism	-	-	end plate in bending	column web in compression	-	end plate in bending	column web in shear	-	
C4	number of web plates	-	-	-	-	-	-	-	0	1 OR 2
A1	advised to modify	height haunch	thickness end plate	height haunch	number of web plates	number of web plates	height haunch	number of web plates	stiffener	height haunch
		R1	R2	R3	R4	R5	R6	R7	R8	R9

Fig. 9 Decision Table System of prototype

4.2 Decision Table *Sufficient Strength*

In DT *Sufficient Strength* the system starts tracing parameter *first moment capacity*. In order to find a value for this parameter the system first executes the WHEN NEEDED demon (see figure 10). This demon represents the subtask *Calculate properties* in the design process (see figure 6). The system is calling CASTA/Connections which returns the first and second moment capacity and the first and second collapse mechanism.

```
PARAMETER     : layout of joint
WHEN NEEDED   : ?thickness of end plate?=5, ?number of web plates?=0, ?height haunch?=0, ?stiffener?=no
TYPE          : set
MAIN GOAL

PARAMETER     : first moment capacity
WHEN NEEDED   : calculate_properties (!thickness of end plate!, !number of web plates!, !height haunch!,
                !stiffener!, ?first moment capacity?, ?first collapse mechanism?, ?second moment capacity?,
                ?second collapse mechanism?)
TYPE          : real

PARAMETER     : element to be modified
PROMPT        : What element do you like to modify? (advised: !advised to modify!)
DEFAULT       : !advised to modify!
TYPE          : text
ASK FIRST
```

Fig. 10 Parameter properties of some parameters

The first moment capacity of the joint indicates the maximum moment which can be transferred through the joint. The occuring mechanism at this moment level is called the first collapse mechanism. It refers to the weakest component in the joint. The second moment capacity is the capacity which can be reached theoretically if the first collapse mechanism is prevented (for example by a stiffener). The occurring collapse mechanism is indicated by the second collapse mechanism. It refers to the one but weakest component in the joint.

If the first moment capacity is less than the required moment capacity the designed connection hasn't sufficient strength yet (parameter *sufficient strength* =no) and the system continues tracing a value for the parameter *layout modified* in DT *Design Layout of Joint.* This parameter doesn't have a WHEN NEEDED demon, so the system looks for a DT having this parameter as action parameter and finds DT *Modify Layout.*

4.3 Decision Table *Modify Layout*

This DT describes which modifications affect the strength of the joint in case of a particular collapse mechanism. This is important knowledge for a designer. Novice designers usually don't have this knowledge. They often apply alternatives which only have limited effect on the strength of the joint. Often stiffeners are applied whatever collapse mechanism occurs.

When the system enters DT *Modify Layout*, parameter *first collapse mechanism* already has a value (calculated by CASTA/Connections; see figure 10: WHEN NEEDED demon of parameter *first moment capacity*), so the first parameter to trace is *element to be modified.* This parameter is defined as ASK FIRST (see figure 10), so the user is asked to provide a value. The PROMPT defined for this parameter (see figure 10) contains a reference to a parameter that hasn't been traced yet: parameter *advised to modify.* This parameter doesn't have a WHEN NEEDED demon, so the system looks for a DT having this parameter as action parameter and finds DT *Modification Advice.*

4.4 Decision Table *Modification Advice*

DT *Modification Advice* gives advice on which of the alternative modifications modelled in DT *Modify Layout* will have most effect on the strength of the joint. The advice primarily depends on the first collapse mechanism, but in some situations also knowledge about the second moment capacity and collapse mechanism influences the advice. DT *Modification Advice*, for instance, describes that in case of end plate in bending as the first collapse mechanism the alternative modifications are applying a haunch (when a haunch is already present, increase its height) or applying a thicker end plate. A novice designer will probably decide to apply a thicker end plate in all situations. However, applying a thicker end plate is only an effective choice, if the second moment capacity exceeds the required moment capacity or the second moment capacity is 1.5 times greater than the first moment capacity. If the second moment capacity is close to the first moment capacity an expert designer knows that applying a thicker end plate will help to increase the first moment capacity of the connection, but it will definitely not be sufficient to reach the required moment capacity. If a novice designer still decides to apply a thicker end plate he will surely need an additional element in the next step. In this situation an expert designer will therefore apply a haunch or increase the height of the present haunch, because a haunch is not only effective to increase the moment capacity if end plate in bending is the collapse mechanism, but is also effective in case of other collapse mechanisms (see DT *Modify Layout.*). By applying a haunch the designer will definitely reach the goal in less steps than by applying a thicker end plate. In many cases it also leads to a cheaper layout, because the novice designer will probably not go back to a thinner end plate again because he has the feeling that he made progress by applying the thicker end plate.

The structure of DT *Modification Advice* shows the importance of functional classifications. All three appearances of functional equivalence are present in this DT: (1) in some situations an additional parameter *second collapse mechanism* or *number of web plates* becomes relevant, (2) depending on the value of the parameter *first collapse mechanism*, parameter *second collapse mechanism* has different categories and (3) the presence of 1 or 2 web plates is equivalent for the advice to modify height haunch in case of column web in compression.

4.5 Continuation of the process

After tracing Decision Table *Modification Advice* the system has a value for the parameter *advised to modify* and shows this value as an advice when prompting the user for a value of the parameter *element to be modified* in DT *Modify Layout*. The user is not forced to follow the advice. The advised element is default (see figure 10), but it's still allowed to choose another alternative. After choosing a value for the parameter *element to be modified*, the system resets the current value of the element and returns to DT *Design Layout of Joint* and executes the Prolog-statement *repeat_table*. DT *Design Layout of Joint* is entered again and a value for the parameter *sufficient strength* is traced followed by a tracing of the parameter *first moment capacity*. As described before a value for this parameter is calculated by executing its WHEN NEEDED demon, i.e. running CASTA/Connections. The difference this time however is that the input parameter, that represents the element to be modified, doesn't have a value yet, because its value was reset in DT *Modify Layout*. So, the system first has to find a value for this parameter before the demon can be executed. Since all parameters representing the joint elements are defined as ASK FIRST (not mentioned in figure 10) the user is asked to provide a value. After giving a value CASTA/Connections calculates new values for the first and second moment capacity and collapse mechanism. The first moment capacity is compared to the required moment capacity etc. This process continues until the first moment capacity is greater than or equal to the required moment capacity. Then parameter *sufficient strength* receives value "yes" and the system halts and reports the final layout.

5. CONCLUSION

This paper described the development of a prototype Knowledge Based System for connection design in steel structures. Although the knowledge represented in the prototype is limited, the approach showed that, following the theory of functional classifications and using the conceptual modelling languages, Decision Tables and Prolog, a practical application can be developed. It turned out that by using Decision Tables and the tool AKTS the expert could easily validate the knowledge and give suggestions for improvement. Consequently this approach will lead to knowledge based systems which better reflect reality and better supports the solution of problems in practice. Hopefully this is an important step towards a situation wherein knowledge based systems will really be accepted as useful tools by people working in the building industry.

6. REFERENCES

1. ECCS, Analysis and Design of Steel Frames with Semi-Rigid Joints, ECCS-TC8, publication no 67, Brussels, Belgium, 1992;
2. EUROCODE 3, ENV - 1993-1-1, Design of Steel Structures, Commission of the European Communities, European Prenorm, Brussels, Belgium, April 1992;
3. BROZZETTI, J., Design of Connections in the EUREKA "CIMSTEEL" project, Proceedings of the Second International Workshop on Connections in Steel Structures, edited by Bjorhovde R., Colson A., Haaijer G. and Stark J., AISC, Chicago, Illinois, USA, 1992;
4. STEENHUIS, M., DOL C. & VAN GORP, L. Computerised calculation of force distributions in bolted end plate connections according to Eurocode 3, to be published in Journal of Constructional Steel Research, 1994;
5. HAYES-ROTH, F. & JACOBSTEIN, N. The State of Knowledge Based Systems, Communications of the ACM, Vol. 37, No. 3, pp. 27-39, March 1994;
6. GELDER, J. T. DE., & LUCARDIE, G. L., Reconstructing Conceptual Models of Structures, Proceeding of IABSE Colloquium Knowledge-Based Systems in Civil Engineering, pp. 65-76, Beijing, China, 1993;
7. LUCARDIE, G.L., A Functional Approach to Realizing Decision Support Systems in Technical Regulation Management for Design and Construction, Design & Decision Support Systems in Architecture (Kluwer Academic Publishers), pp. 97-109, 1993;
8. GELDER, J. T. DE., & LUCARDIE, G. L., Knowledge and Data Modelling for CAD/CAM Applications, Design & Decision Support Systems in Architecture (Kluwer Academic Publishers), pp. 111-121, 1993;
9. LUCARDIE, G.L., GELDER, J. T. DE, HUIJSING, A.P., DTSS: A Functional Application of Decision Tables and Prolog as a Conceptual Modelling Language for Knowledge Based Systems, Proceedings of the 1st Workshop of the European Group for Structural Engineering Applications of Artificial Intelligence (EG-SEA-AI), pp. 192-204, Lausanne, Switzerland, 1993;
10. FENVES, S.J., Tabular Decision Logic for Structural Design, Journal of the Structural Division, 92(6), pp. 473-490, 1966;
11. GARRETT, J.H.& HAKIM, M.M., Object-Oriented Model of Engineering Design Standards, Journal Computing in Civil Engineering, Vol. 6, No. 3, pp. 323-347, 1992;
12. GARRETT, J.H.& FENVES, S.J., A Knowledge-Based Standards Processor for Structural Component Design, Engineering with Computers, Vol. 2, No. 2, pp. 219-238, 1987;
13. GARRETT, J.H.& FENVES, S.J., Knowledge-Based Standard Independent Member Design, Journal of Structural Engineering, Vol. 115, No. 6, pp. 1396-1411, 1989;
14. CRONEMBOLD, J.R. & LAW, K.H., Automatic Processing of Design Standards, Journal Computing in Civil Engineering, Vol. 2, No. 3, pp. 255-273, 1988.

Leere Seite
Blank page
Page vide