KNOWLEDGE AND DATA MODELLING IN CAD/CAM APPLICATIONS

J.T. DE GELDER and G.L. LUCARDIE
TNO Building and Construction Research
P.O. Box 49
2600 AA Delft
The Netherlands

ABSTRACT. Modelling knowledge and data in CAD/CAM applications is complex because different goals and contexts have to be taken into account. This complexity makes particular demands upon representation formalisms. Today many modelling tools are based on record structures. By analyzing the requirements for a product model of a portal structure in steel, this paper shows that in many situations record structures are not well suited as a representation formalism for storing knowledge and data in CAD/CAM applications. This is illustrated by performing a knowledge-level analysis of the knowledge and data generated in the design and manufacturing process of a portal structure in steel.

1. Introduction

A large number of DataBase Management Systems (DBMS) applied in many commercial applications are implementations of record-based models (Date 1985). Record structures are mainly successful because of their efficiency and ease of manipulation. Record structures are excellent data-processing tools in application areas where objects have identical attributes, attributes always have the same type of value, and relationships between objects are relatively static. These application areas are called homogeneous. However, record structures are not very well suited for so-called heterogeneous application areas. In such cases, objects sometimes have to be described by many different and frequently changing attributes. Moreover, attributes can have different types of values, changing interactions between attributes can exist, and relationships between objects can be dynamic.

This paper performs a knowledge-level analysis of the knowledge and data generated in the design and manufacturing process of a portal structure in steel. This analysis demonstrates that CAD/CAM knowledge and data is heterogeneous, implying that CAD/CAM applications belong to the area of heterogeneous applications. The knowledge-level analysis has been conducted using the theory of functional classifications (Van der Smagt 1985; Hendriks 1986; Reitsma 1990; Van der Smagt and Lucardie 1991). This paper is organized as follows. First, the knowledge-level analysis itself is described in Section 2. Section 3 then describes an evaluation of record structures as a representation formalism for storing CAD/CAM knowledge and data. Next, in Section 4, some conclusions and alternative research directions are presented.

2. A Knowledge-Level Analysis

In his presidential address to the American Association of Artificial Intelligence (AAAI), Newell introduced a new modelling level: the knowledge level (Newell 1981). At this level, knowledge about an application area can be defined (Newell 1981; Brachman and Levesque 1986; Berg-Cross and Price 1989; Twine 1989). A knowledge-level analysis of an application area permits the development of implementation-independent conceptual (product) models, which

H. Timmermans (ed.), Design and Decision Support Systems in Architecture, 111–121.

better reflect the reality of an application area (Levesque 1984; Brachman and Levesque 1986; Steels 1990). These models can then be used to evaluate representation formalisms like record structures to see whether they can be used to store the knowledge and data of the conceptual model (Brachman and Levesque 1986).

In order to perform a knowledge-level analysis, a theory has to be adopted that prescribes how the analysis should be performed. In this paper, the theory of functional classifications is used (Van der Smagt 1985; Hendriks 1986; Reitsma 1990; Van der Smagt and Lucardie 1991). The theory of functional classifications considers knowledge as a set of concepts. Each concept has an intension and an extension. The intension of a concept is a set of attributes and constraints, also called an object-type. The extension of a concept is a set of objects satisfying the intension. Several theories deal with the modelling of object-types. The theory of functional classifications, however, is unique in the way the set of attributes and constraints is reconstructed. In reconstructing object-types, two aspects are essential: the functional aspect and the relational aspect.

2.1. THE FUNCTIONAL ASPECT

A key element in the theory of functional classifications is that the reconstruction of object-types is always governed by a specific function or goal. This means that the reconstruction of an object-type always underlies a certain function. For instance, the function of a connection in a portal structure in steel is: joining steel profiles. With respect to this function, it is conceivable that mechanical requirements determine the reconstruction of the intension of the concept 'connection'. All objects meeting these requirements are part of the extension of the concept 'connection'. It is possible that objects satisfying the intension are quite different. Notwithstanding the differences, these objects are functionally equivalent. Functional equivalence of objects is a main element in the theory of functional classifications. Objects which at first sight seem to differ may be equivalent if we consider the function they have to fulfil. The occurrence of different objects which appear to be functionally equivalent will be demonstrated by a simplified example of the design process of a portal structure in steel.

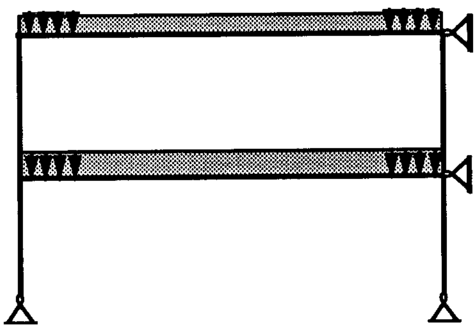

Figure 1. Portal structure with loads

In the design process of a portal structure in steel, three main activities can be recognized (Steenhuis and Dol 1991): design of the portal structure and the loads, design of the profiles, and design of the connections. The starting point is the portal structure displayed in Figure 1. The portal structure consists of two columns and two beams. Figure 1 also shows the loads on the two beams. The portal structure and loads are the basis for designing the profiles. The required profile types for the columns and the beams can be calculated. Some results of this calculation are, for instance, that profile type HE220A is satisfactory for both columns, profile type IPE220 is satisfactory for the roof beam, and profile type IPE330 for the floor beam. Figure 2 displays the portal structure again, but now after the design of the profiles. Since no connections have been designed yet, the profiles overlap at the place of the nodes. After designing the profiles, the required moment capacity in the nodes can be calculated. The moment capacity is an indication of the strength of a connection. Based on the results of this calculation, connections are designed at each node to meet the required moment capacity. We will concentrate on node 2 and design an end plate connection for it. In column-to-beam end-plate connections, the actual joint between the column and the beam is provided by an end plate that is welded to the beam and bolted to the column.

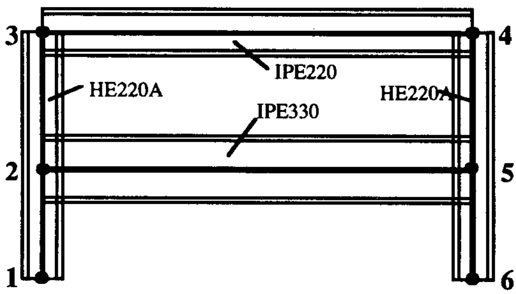

Figure 2. Portal structure with profiles

Above, the function of a connection in a portal structure was defined as: joining steel profiles. For the particular connection in node 2, a more specific function can be defined namely: connecting a beam (profile type IPE330) to a column (profile type HE220A) by means of an end plate. The required moment capacity in node 2 governs the reconstruction of the intension of the concept 'column-to-beam end-plate connection'. Based on the profile design, we found, for instance, that the moment capacity should be at least 60 kNm. The extension of the concept 'column-to-beam end-plate connection' is the set of all column-to-beam end-plate connections between an HE220A and an IPE330, having a moment capacity of at least 60 kNm. In practice, it is possible to design several different column-to-beam end-plate connections having a moment capacity of at least 60 kNm.

Figure 3a displays the first connection design including an end plate with a thickness of 5 mm. The moment capacity appears to be insufficient (25kNm), and the determining collapse mechanism is yielding of the end plate. This means that if the moment capacity exceeds a value of 25kNm, the first element in the connection that will collapse is the end plate. In a situation where yielding of the end plate is the determining collapse mechanism, there are two alternatives to increase the moment capacity of the connection: (i) including a thicker end plate to increase the strength of the end plate, which is the weakest element of the connection, and (ii) including a haunch to spread the moment force over a longer distance and so reduce the moment force on the end plate.

If the end plate, with a thickness of 5 mm, is replaced by an end plate with a thickness of 10 mm (Figure 3b), the moment capacity is still insufficient (56kNm) and the determining collapse mechanism is changed into shear of the column web. This means that if the moment capacity exceeds a value of 56kNm, the first element in the connection that will collapse is the column web. In a situation where shear of the column web is the determining collapse mechanism, including an end plate thicker than 10 mm makes no sense. The moment capacity of the connection in Figure 3a can also be increased by including a haunch. It was found that by including a haunch with a height of 330 mm, the moment capacity can be increased to 60kNm. The determining collapse mechanism remains yielding of the end plate. Figure 3c displays the connection when a haunch with a height of 330 mm is applied.

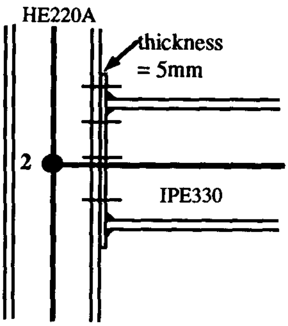

Figure 3a. Moment capacity = 25kNm

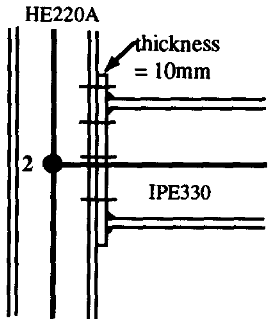

Figure 3b. Moment capacity = 56kNm

The solution in Figure 3c meets the requirements, since the moment capacity should be at least 60kNm. The number of bolt rows has to be raised from four to seven because of the haunch. The moment capacity of the connection in Figure 3b appeared to be not sufficient yet. Shear of the column web was found to be the determining collapse mechanism of this connection. Considering this collapse mechanism, there are two alternatives to increase the moment capacity of the connection: (i) including a web plate to strengthen the column web, which is the weakest element of the connection, and (ii) including a haunch to spread the moment force over a longer distance and so reduce the moment force on the column web. Figure 3d displays the connection after including a haunch with a height of 60 mm.

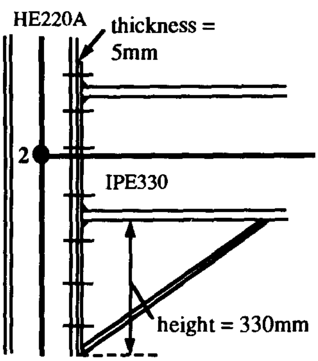

Figure 3c. Moment capacity = 60kNm

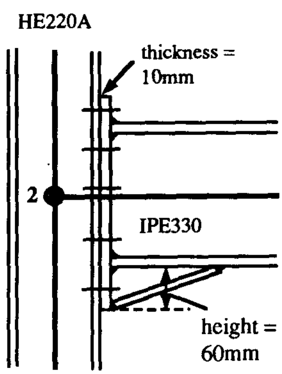

Figure 3d. Moment capacity = 66kNm

The moment capacity of this connection is 66kNm, and the determining collapse mechanism remains shear of the column web.The solution in Figure 3d meets the requirements, since the moment capacity should be at least 60kNm. The number of bolt rows has to be raised from four to five because of the haunch. If, instead of including a haunch, a column web with a thickness of 7 mm is included, the moment capacity turns out to be 82kNm. The determining collapse mechanism remains shear of the column web. Figure 3e displays the connection with a column web applied.

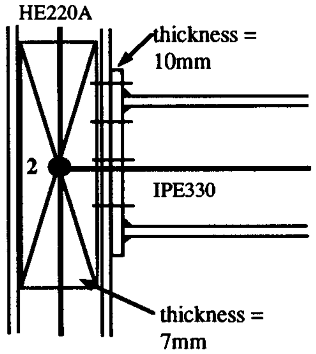

Figure 3e. Moment capacity = 82kNm

Figure 4 describes the connection design process by means of a disjunction of conjunct sets. Only the main attributes are included to this model. These conjunct sets all lead to the same goal: required moment capacity >= 60 kNm. Every element of a conjunct set is an *INUS-condition* (Denise 1984): an Insufficient but Necessary part of a set which is Unnecessary but Sufficient for the result. Within a conjunct set, an *INUS-condition* is indispensable for achieving a goal. But theconjunct set to which the *INUS-condition* belongs is replaceable by other conjunct sets. The conjunct sets in Figure 4 are:

1. {(end plate;[5,..]),(haunch;[330,..]), (bolt;[7,16,8.8,..])},
2. {(end plate;[10,..]),(web plate;[7,..]), (bolt;[4,16,8.8,..])}
3. {(end plate;[10,..]),(haunch;[60,..]), (bolt;[5,16,8.8,..])}

For example, *INUS-condition* (haunch;[330,..]) is an Insufficient part of conjunct set 1, because the *INUS-conditions* (end plate;[5,..]) and (bolt;[7,16,8.8,..]) are necessary too. However, *INUS-condition* (haunch:[330,..]) is Necessary for conjunct set 1, because a conjunct set only consisting of *INUS-conditions* (end plate:(5,..)) and (bolt:(7,16,8.8,..)) is insufficient for reaching the goal (see Figure 3a). Conjunct set 1 is Unnecessary for reaching the goal, because the goal can also be reached by conjunct set 2 or conjunct set 3. However, conjunct set 1 is Sufficient for reaching the goal. An object must satisfy one of the conjunct sets in order to be classified as a column-to-beam end-plate connection between an IPE330 beam and an HE220A column with moment capacity >= 60 kNm.

This example demonstrates that geometrically completely different objects are functional equivalents if the function is: connecting a beam member (profile type IPE330) to a column member (profile type HE220A) by means of an end plate such that the moment capacity >= 60 kNm. This example also demonstrates that the set of attributes and constraints forming the intension of the concept 'column-to-beam end-plate connection' is rather heterogeneous. The object-type 'column-to-beam end-plate connection' has many different and frequently changing attributes, which are actually object-types in themselves, since they also have their own attributes.

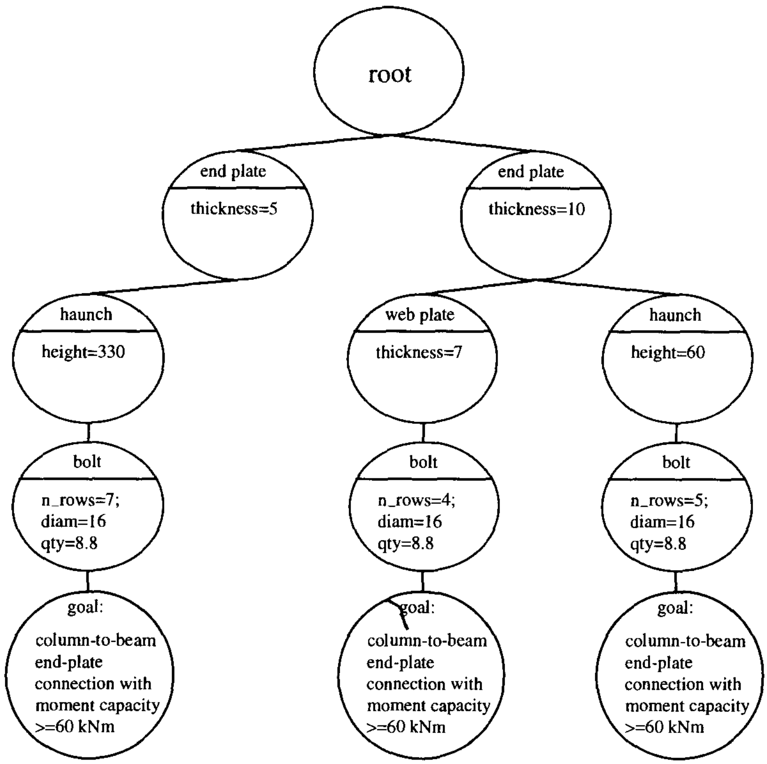

Figure 4. A disjunction of conjunct sets

2.2. THE RELATIONAL ASPECT

Another element in the theory of functional classifications is that classifications of object-types are only valid within one specific context. The intension of a concept is governed not only by a function or goal, but also by a context. This is demonstrated by comparing two phases in realizing a portal structure in steel: the design phase and the manufacturing phase. These phases can be considered as two different contexts. In the design phase, a set of mechanical requirements governs the reconstruction of the intension of the concept 'portal structure'. In order to meet these requirements, the designer designs profiles and connections between profiles (Figure 5).

In the manufacturing phase, a set of economical requirements governs the reconstruction of the intension of the object-type 'portal structure'. In order to meet these requirements the manufacturer produces assemblies in house, which are connected on site. Generally an assembly is a profile including all the parts welded to the profile. Because of a different context, a manufacturer describes the portal structure in a completely different way (Figure 6).

A different description means different attributes of object-types and different relationships between object-types. For example, in the design context, the end plate is part of the column-to-beam end-plate connection, in the manufacturing context, the end plate is part of the beam assembly. Likewise, in the design context, holes are not defined explicitly, because the existence of a hole is implicitly defined by the existence of a bolt. In the manufacturing context, a hole has to be defined explicitly and related to the column profile and the end plate, because holes have to be drilled into these parts in the manufacturing phase. Similarly, in the design context, the end

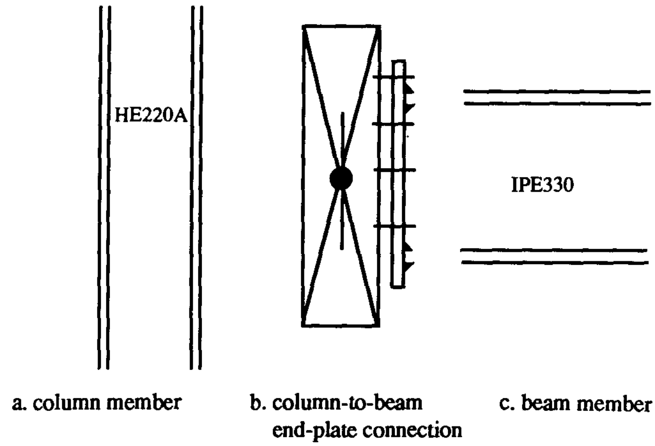

Figure 5. Design context

plates, haunches and web plates are located relative to the connection node. In the manufacturing context, these parts are located relative to either one end of the beam profile or one end of the column profile. Also, in the design context, a number of part dimensions are only implicitly defined, since they are mechanically unimportant or dependent on other dimensions. The length of the beam profile, for instance, is partly dependent on the height of the column and the thickness of the end plate. In the manufacturing context, all dimensions are explicitly defined, because parts have to be manufactured.

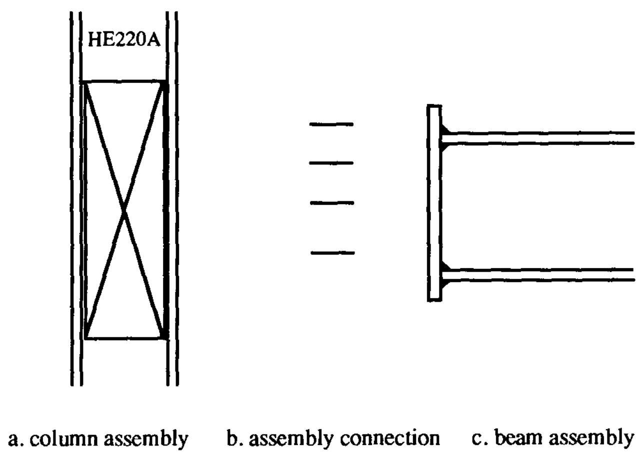

Figure 6. Manufacturing context

2.3. COMMON PRACTICE IN PRODUCT MODELLING

In product modelling practice, the functional and relational aspect in reconstructing the intension of a concept is often not recognized. The general problems that can be found in product modelling are no clear distinction of levels, no clear distinction of goals, and mixing of contexts. Product models are often developed using modelling techniques which make no clear distinction between knowledge level and symbol level. Often knowledge of reality is adapted to the limitations of representation formalism. The functional aspect stated that the reconstruction of object-types is always governed by a specific function or goal. In practice, product models are often developed without a specific modelling goal in mind. One just starts modelling a product without considering the goal of modelling. Since the reconstruction of object-types is always governed by a specific goal, modelling without a specific goal leads to general models, which are not useful for specific purposes. The relational aspect stated that the classification of object-types is only valid within one specific context. In the development of a product model for integrated CAD/CAM applications, often only one classification is reconstructed for all contexts. Either a classification for one of the contexts is selected for the product model or a mixture of all required classifications is developed for the product model. Again this leads to general models, which are not useful in a specific context.

3. Limitations of Record-Based Models

At the symbol level, object-types, which are reconstructed at the knowledge level, are reduced to representation formalisms. These consist of structures to store knowledge and data as well as structures to access mechanisms in order to operate on knowledge and data. For storing and accessing knowledge and data, record-based models are often applied, especially so-called relational models. In a relational model, object-types are stored as rather simple record structures. The intension is defined by a fixed number of fields, each field having a predefined value type. The extension consists of a number of records. A record is a fixed number of values, each corresponding with a field of the intension. Each value must be part of the range defined by the value type of the corresponding field. Value types are generally restricted to numeric and character string data types. Figure 7 shows an example of records in a relational model.

CONNECTION:

con.id	end plate.id	haunch. id	web plate.id	moment-capacity
01	11	21	NULL	60
02	12	NULL	31	82
03	12	22	NULL	66

END PLATE:

end plate.id	thick-ness
11	5
12	10

HAUNCH:

haunch. id	end plate.id
21	330
22	60

WEB PLATE:

web plate.id	thick-ness
31	7

Figure 7. Record structures in a relational model

The boxes in Figure 7 are called tables. Relationships between records can be defined by accessing the records and searching for identical attribute values. By accessing, for instance, the first record of an object-type 'connection', the related record of object-type 'end plate' can be found by searching the set of records for the attribute value '11' of the common attribute 'end plate.id'. How can the object-types, reconstructed at the knowledge level for the portal structure in steel, be stored by record structures at the symbol level? In Section 2.1 it was found that in CAD/CAM applications, functionally equivalent objects can be geometrically different. It was found, too, that these functional equivalent objects are complex objects consisting of frequently changing sub-objects. Modelling complex object-types in a relational model is difficult, because the access mechanism of a relational model is not able to deal with hierarchical ('part of') relationships. In a hierarchical relationship, one object-type is existence-dependent on another object-type. In Figure 7, end plate '11' is existence-dependent on connection '01'. If connection '01' is deleted from the model, the model should react automatically by deleting end plate '11' and haunch '21' as well. The relational model, however, would react to this request by deleting only connection '01' and leaving end plate '11' and haunch '21' untouched.

Functionally equivalent objects consist of frequently changing sub-objects. In Figure 4, the object 'column-to-beam end-plate connection' is at one time related to the sub-object 'web plate' and another time to the sub-object 'haunch'. In a relational model, relationships are defined by means of common attributes, so one time the object-type 'column-to-beam end-plate connection' has a common attribute with the object-type 'web plate' and another time with the object-type 'haunch'. It is possible to solve this problem of different attributes by including supertype/subtype relationships between object-types into the relational model. A supertype and its subtypes are functionally equivalent object-types. The supertype holds the common attributes of the functional equivalent object-types. A subtype holds the unique attributes, which differ from attributes of the other object-types. In the example, the supertype 'connection part' could be defined, having two subtypes 'haunch' and 'web plate'. The relationship between the supertype 'connection part' and its subtypes 'haunch' and 'web plate' defines that a connection part is either a haunch or a web plate. The problem of changing relationships between the object-type 'column-to-beam end-plate connection' and either the object-type 'web plate' or the object-type 'haunch' can be solved then by defining a fixed relationship between the object-type 'column-to-beam end-plate connection' and the object-type 'connection part'. This solution of defining a supertype/subtype relationship between on the one hand the object-type 'connection part' and on the other hand the object-types 'haunch' and 'web plate' is only appropriate when the objects 'haunch' and 'web plate' are functionally equivalent in any situation. In Section 2.1, however, it was demonstrated that in case of a determining collapse mechanism indicated as yielding of the end plate, haunch and web plate are not functionally equivalent. Thus, in that case they cannot be represented as two subtypes of the same supertype.

In Section 2.2, it was found that attributes of object-types and relationships between object-types in CAD/CAM applications can change dramatically after changing the context from design to manufacturing. In Kent (1979), several possibilities for modelling changing attributes are presented, all of which have their drawbacks. For all attributes of an object-type required in any context, a field can be defined. The drawback of this so called *multiple fields* solution is that for a particular context, only part of the attributes is relevant and the rest should be assigned null values. Another possibility is to create *multi-functional fields.,* indicating that an attribute of an object-type can have different meanings, depending on the context. The drawback of this solution is that such a practice is never defined to the system. With respect to any processing done by the system, that field appears to have the same significance in every record occurrence. The different meanings these fields have is only known in application programs. Finally, *multiple object-types*, suggesting that an object-type can be spread over several tables. Each table stores a set of attributes for one specific context. The drawback of this solution is that one object-type is split into different object-types. These are not real object-types concerning the reality of the application area, but only created because of the limitations of the representation formalism.

4. Conclusion

Record-based models are excellent modelling tools for application domains in which data and knowledge have a homogeneous nature. Both object-types and relationships should be homogeneous. This paper showed that knowledge and data in a CAD/CAM domain can hardly be represented by records, because of the heterogeneous nature of knowledge and data in both object-types and relationships. This has been demonstrated by performing a knowledge-level analysis of the knowledge and data generated in the design and manufacturing process of a portal structure in steel.

Since record-based models appear not be suitable for representing knowledge and data in a CAD/CAM domain, a lot of research has been conducted on alternative representation formalisms. These have more flexibility, as well as the dynamics to represent knowledge and data in heterogeneous application areas. This research has not only been conducted in the field of database technology (Ter Bekke 1983; Hull and King 1987), but also in the field of Artificial Intelligence (Loucopoulos and Karakostas 1989).

Recently, research has been started on the integration of knowledge bases and databases (Lucardie 1992). An important aspect of this research is that integration takes place at the knowledge level. Conceptual (product) models have to developed independent of any implementation strategy. After developing these knowledge-level models, attention will have to be paid to representation formalisms to represent these models at the symbol level. Hopefully the research on alternative representation formalisms will lead to representation formalisms which are able to represent all knowledge and data reconstructed at the knowledge level without reducing the semantics and dynamics of this knowledge and data. If so, the gap between knowledge-level and symbol-level models can be closed. Then information systems can be developed which better meet the requirements of the application domain.

References

Bekke, J.H. ter (1983) *Database Ontwerp*, H.E. Stenfert Kroese, Leiden-Antwerpen.

Berg-Cross, G. and Price, M.E. (1989) "Acquiring and Managing Knowledge Using a Conceptual Structures Approach: Introduction and Framework", *IEEE Transactions on Systems, Man and Cybernetics* **19**, 513-527.

Brachman, R.J. and Levesque, H.J. (1986) "What Makes a Knowledge Base Knowledgeable? A View of Databases from the Knowledge Level", in L. Kerschberg (ed.), *Expert Database Systems*, Benjamin/Cummings, Menlo Park, pp. 69-78.

Date, C. (1985) *Database: Een Inleiding*, Academic Service, Den Haag.

Denise, T.C. (1984) "On the Nature of INUS Conditionality", *Analysis* **44**, 49-52.

Gelder, J.T. de (1993) Design of a Central Database as Part of an Integrated System for Computer Aided Design of Steel Structures, TNO Building and Construction Research, report no. BI-88-076.

Hendriks, P.H.J. (1986) De Relationele Definitie van Begrippen: Een Relationeel Realistische Visie op het Operationaliseren en Representeren van Begrippen, Ph.D. Dissertation, Faculty of Policy Sciences, University of Nijmegen.

Hull, R. and King, R. (1987) "Semantic Database Modeling: Survey, Applications and Research Issues", *ACM Computing Surveys* **19**, 201-260.

Kent, W. (1979) "Limitations of Record-Based Information Models", *ACM Transactions on Database Systems* **4**, 107-131.

Levesque, H.J. (1984) "Foundations of a Functional Approach to Knowledge Representation", *Artificial Intelligence* **23**, 155-212.

Loucopoulos, P. and Karakostas, V. (1989) "Modeling and Validating Office Information Systems: An Object and Logic Oriented Approach", *Software Engineering Journal* **4**, 87-94.

Lucardie, G.L. (1992) "A Functional Approach to Realizing Decision Support Systems in Technical Regulation Management for Design and Construction", in H. Timmermans (ed.), *Design and Decision Support Systems in Architecture*, Kluwer, Dordrecht.

Newell, A. (1981) "The Knowledge Level", *AI Magazine* **2**, 1-20.

Reitsma, R.F. (1990) Functional Classification of Space: Aspects of Site Suitability Assessment in a Decision Support Environment, Ph.D. Dissertation, International Institute for Applied Systems Analysis, Laxenburg, Austria / Faculty of Policy Sciences, University of Nijmegen.

Smagt, A.G.M. van der (1985) Definiëren en Relateren in Sociaal-Wetenschappelijk Onderzoek, Ph.D. Dissertation, Faculty of Policy Sciences, University of Nijmegen.

Smagt, A.G.M. van der and Lucardie, G.L. (1991) "Decision-Making Under Not-Well-Defined Conditions: From Data Processing to Logical Modelling", *Tijdschrift voor Economische en Sociale Geografie* **82**, 295-304.

Steels, L. (1990) "Components of Expertise", *AI Magazine* **11**, 29-49.

Steenhuis, C.M. and Dol, C. (1991) "Geboute Kopplaatverbindingen: Gebruik van NEN 6772 tijdens het Ontwerpproces van een Staalconstructie", *Bouwen met Staal* **103**, 3-7.

Twine, S. (1989) "Mapping Between a NIAM Conceptual Schema and KEE Frames", *Data & Knowledge Engineering* **4**, 125-155.