Knowledge eliciting and modelling using the entity relationship approach

Spoor, E.

1986

document version
Publisher's PDF, also known as Version of record

citation for published version (APA)
Spoor, E. (1986). *Knowledge eliciting and modelling using the entity relationship approach*. (Serie Research Memoranda; No. 1986-21). Faculty of Economics and Business Administration, Vrije Universiteit Amsterdam.

SERIE RESEARCH MEMORANDA

KNOWLEDGE ELICITING AND MODELLING USING THE ENTITY RELATIONSHIP APPROACH

a graphical language to build prototype knowledge based systems

E.R.K. Spoor

Researchmemorandum 1986-21
juni 1986

VRIJE UNIVERSITEIT
FACULTEIT DER ECONOMISCHE WETENSCHAPPEN
AMSTERDAM

ABSTRACT

The design of knowledge based systems draws considerable attention from various scientific disciplines. But researchers merely focus on three central parts: the linguistic part, the problem solving techniques and the representation of knowledge.

Few attention has been payed to the process of eliciting relevant knowledge and the formal reconstruction of it by means of model building techniques. In this paper an extended form of the Entity Relationship Approach[1] is presented as a possible connection between psychological methods for revealing the experts knowledge at one side and implementation techniques at the other.

1. INTRODUCTION

To obtain a picture of what is 'known' by an expert, one could interview him or her and write an informal note on that interview. This note than serves as a representation of (some of) his or her knowledge.

A formal system which acts like an expert (a point we try to reach building knowledge based systems) needs a formal representation of the knowledge. Questions arise when speaking of knowledge representation. What is knowledge and what do we mean by representation? Shefe[2] recognizes several levels of modelling, among which the formal and the informal representation levels. Each level constitutes the body of the next higher level. Only at the lower formal levels there is always a one-to-one 'interpretation'. The difficulties arise between:

1) the level of concepts 'known' by the expert and the level of informal representation by those concepts;
2) the level of informal representation and the level of formal representation.

Phenomenons like homonyms and synonyms stress the problem of constituting the body of the informal representation level from the formal representation level. Likewise, informal representations (words) do not constitute fully the body of a known concept. The concept is also known by much fuzzy background knowledge. So we are actually dealing with the 'formal reconstruction of knowledge and its implementation'[3] when we speak of knowledge representation.

There is not much knowledge about knowledge. A general notion is that of perception from phenomena which results in abstract concepts referring to those phenomena[4]. In this view expert knowledge is the result of a perception process restricted to some domain of phenomena.

In a recent article Newell[5] introduces a new formal systems level immediately above the formal representation level: the knowledge level.

Being fully constructed from the formal level below, the knowledge level does not suffer from the aforementioned problems like homonyms or synonyms because its underlying level is a formal level. Newell draws several conclusions, three of which are recalled here:

1) knowledge is that which makes the principle of rationality work as a law of behaviour. Thus, knowledge and rationality are intimately tied together;
2) knowledge is not representable by a structure at the symbol level. It requires both structures and processes. Knowledge remains for ever abstract and can never be actually in hand;

3) knowledge is a radical approximation, failing on many occasions to be an adequate model of an agent. It must be coupled with some symbol level representation to make a viable view.

So structures and processes at the formal representation (symbol) level realize a body of knowledge at the knowledge level.

To construct a knowledge based system which 'resembles the performance' of the expert, one must determine what datastructures are needed and which processes can be defined to realize an adequate body of knowledge.

Since an expert, like all human beings, uses the informal representation level as a way to communicate, and since his or her knowledge can never be completely described by this representation, the formal reconstruction of human knowledge must lead to the creation of an incomplete and probably unrealistic formal model. This formal model serves as a heuristic device[6] to realize some body of knowledge.

Though disputable, graphical languages often serve as a medium between expert and modeller to reduce the level transformation problems. They never provide a complete picture of the problem area but merely serve as a tool for understanding the domain. The implementation of a graphical model can best be considered as a prototype, subject to subsequent improvements.

In this paper a formal treatment of an extension of the Entity Relationship Approach[1] as a graphical language for prototype building, is presented (section 3.1 and 3.2). Furthermore a mapping is defined to transform the language into well-formed formulas for first-order logic oriented implementation of the prototypes. But first, the connection of the approach to eliciting and analysis techniques is briefly discussed in the sections 2.1 and 2.2.

2. THE INFORMAL PART

2.1 Eliciting techniques

The first level-transformation, i.e. from the knowledge level into the informal representation level, is concerned with the description of what is known by the expert. His knowledge has a diverse nature. He knows of concepts and relations, but also of facts and heuristics. He is able to classify similar concepts and apply routine techniques. To reveal this knowledge we need an eliciting method which covers all aspects. None of the available methods like interview techniques, protocol analysis, etc., cover the whole range. They all suffer from limitations and focus on only some of the aspects.
Gammack & Young[7] suggest the use of several psychological methods to reveal as much as possible of the knowledge just because of its diverse nature. Interviews (and lectures) provide a general impression of concepts and facts, while protocol analysis highlights the knowledge of procedures, heuristics and facts. Multidimensional scaling and concept-sorting focus on revealing a classification of concepts and relations. The result of the application of these methods is an (incomplete) informal description of the experts' knowledge, written in some natural language.

2.2 Analysis

In order to achieve a formal reconstruction of the expert's knowledge, transformation of the informal description into a comprehensive, unambiguous one is the necessary next step.
Several suggestions towards this process have been done[8]: reduction of synonymical expressions into one; resolving homonyms into suitable substitutions; changing pronouns into nouns; etc., all leading to the observation that analysis can not only be a linear comprehension process, but moreover must be an interactive process (i.e. interaction with the expert).
The 'filtered'[9] description enables us to indicate surface markers to distinguish from sentences describing states and those describing state transitions. To be more precise: we consider a filtered description to be a collection of situation descriptions[10] where situations comprise occurrences and states, and each occurrence is either a process or an event[11].
Roland[12], Tsichritzis[13] and de Antonellis[14] use a similar convention: object sentences describe the states, operation sentences describe the transitions from states to states, and event sentences describe the conditions under which transitions occur.
Several checks are recommended at this stage of design to prevent from

unnecessary incompleteness and inconsistency errors[15].
Ultimately we end up with three classes of sentences expressing information on:

1) the properties of data items, associations between those items and classes of data items (the object sentences);
2) procedures to be performed on data (the operation sentences);
3) conditions and consequences (the event sentences).

3. THE FORMAL PART

3.1 Modelling objects

3.1.1 Basic Entity-Relationship (ER)

The data items mentioned in the previous chapter are called entities in the Entity-Relationship Approach[16]. An entity symbolizes a concept. We may classify entities, given a certain context, into sets by means of a function[17] TYPE : $\{e \mid e \text{ is an entity}\} \rightarrow \eta$. The elements of η are type names (N). Two entities e_1 and e_2 are of the same type if TYPE (e_1) = TYPE (e_2).
So the function TYPE assigns a name to each set of entities. If N is a typename, then δN denotes its matching entity set.
Let $r=(e_1,\ldots,e_n)$ be a relationship between the entities $e_1,\ldots,e_n$. Relationships may be classified in the same manner as entities by extending the function TYPE:
TYPE : $\{e \mid e \text{ is an entity}\} \cup \{r \mid r \text{ is a relationship}\} \rightarrow \eta$, with additional constraints:

1) r_1 and r_2 are of the same type if TYPE (r_1) = TYPE (r_2);
2) if TYPE (r_1) = TYPE (r_2) then r_1 and r_2 are defined on the same set of entities;
3) $\forall\, e,r$: TYPE $(e) \neq$ TYPE (r).

Properties imputed to entity sets, named attributes, are viewed as functions from entity sets into value sets. Let δN denote an entity set. An attribute a defined on δN is a function from δN into a valueset: $a\colon \delta N \rightarrow V(a)$.
The identifier of δN is defined to be the minimal set of attributes which gives a one-to-one mapping from δN into the Cartesian product of associated value sets[18].
An entity set δN together with his attributes $a_1,\ldots.a_n$ of which $\{a_1,\ldots,a_k\}$ is the identifier $(k \leq n)$ is characterized by $N(\underline{a_1,\ldots,a_k},a_{k+1},\ldots,a_n)$. Thus an entity set is designated by its associated name and its attribute functions.
A relationshipset δN_r with attributes $b_1,\ldots,b_m$, defined on $\delta N_1,\ldots,\delta N_n$, is characterized as $N_r(N_1,\ldots,N_n\colon b_1,\ldots,b_m)$.
So relationshipssets are designated by their name, their attribute functions and their associated entity sets.

3.1.2 First extension: aggregation

Generally speaking, by aggregation we mean treating a collection of properties as one single property.
Several definitions of it appeared since the introduction of the concept[19]. We stick to the original Cartesian aggregation[20] where a relationship between entities is regarded as a new entity. The reason for choosing the cartesian aggregation is given by its diagrammatic simplicity (see figure 1, where we follow the notation of Webre[21]) and its uniform notation as will be seen hereafter.

figure 1

Formally, let U be the set of all entities and relationships in the UoD. We extend U as follows: Let δN_r be defined on $\delta N_1,\ldots,\delta N_n$. Every element of δN_r is considered to be an entity and as such added to U. This results in a new entity set to which the function TYPE assigns a name, say N^*_r. So there exists a bijective function
$\xi\colon \delta N_r \to \delta N^*_r$.

Let $\{a^i_k,\ldots,a^i_k\}$ be the identifier of δN_i $(1 \leq i \leq n)$.

To δN^*_r we add the following attribute function:

1) for each a^i_j $(1 \leq i \leq n,\ 1 \leq j \leq k)$ of δN_r add a^{*i}_j so that:

$\forall r \in \delta N_r\ ,\ \exists r^* \in \delta N^*_r : \xi(r) = r^* \wedge a^i_j(r) = a^{*i}_j(r^*)$.

2) for each b_i $(1 \leq i \leq m)$ of δN_r add b^*_i so that:

$\forall r \in \delta N_r\ ,\ \exists r^* \in \delta N^*_r : \xi(r) = r^* \wedge b_i(r) = b^*_i\ (r^*)$

Since the new attributes a^{*i}_j and b^*_i can be considered as extensions of a^i_j and b_i respectively[22], we may use the same attribute names, omitting the asterics.

The relationshipset δN_r can be seen as $N^*_r(\underline{a^1,\ldots\ldots,a^n},b_1,\ldots\ldots,b_m)$ whereas N^*_r is the name of the aggregate, $\{a^1,\ldots\ldots,a^n\}$ constitutes the identifier (each $a^i = \{a^i_1,\ldots\ldots,\ a^i_k\}$ is identifier of $\delta N_i (1 \leq i \leq n)$) and $b_1,\ldots\ldots b_m$ are the non-key attributes of the aggregate.

3.1.3 Second extension: generalization

Generalization is most often treated as an abstraction in which a collection of similar entitysets is regarded as a generic set[23]. The similarity depends on the attributes the entity sets have in common. This generalization is exclusive in the sense that an entity belonging to the generic set also belongs to exactly one of the subsets from which is generalized. The exclusiveness is guaranteed by the 'underlying' attribute who's values designate the subset to which each entity belongs[24].
Let δN_1 and δN_2 be disjunct entitysets ($\delta N_1 \cap \delta N_1 = \phi$).
We add to δN_1 and δN_2 the following attributes:
1) to both: $a_i : \delta N_1 \cup \delta N_2 \to V(a_i)$ $(i = 1,\ldots,\ell)$
2) to δN_1 : $a^1_i : \delta N_1 \to V(a^1_i)$ $(i = \ell+1,\ldots,n)$
3) to δN_2 : $a^2_i : \delta N_2 \to V(a^2_i)$ $(i = \ell+1,\ldots,m)$
Which means that δN_1 and δN_2 have the attributes a_i $(i=1,\ldots,\ell)$ in common. Let $\{a_1,\ldots,a_k\}$ be the identifier of δN_1 and δN_2 $(k \leq \ell)$.
The <u>generalization</u> δNg of δN_1 and δN_2 is their union ($\delta Ng = \delta N_1 \cup \delta N_2$).
δNg has the following properties:
1) a_i $(i=1,\ldots,\ell)$, a consequence of the fact that $\delta Ng = \delta N_1 \cup \delta N_2$.
2) $c: \delta Ng \to \{N_1, N_2\}$ where c denotes a category attribute
   $c(e) = N_1$ if $e \in \delta N_1$
   $c(e) = N_2$ if $e \in \delta N_2$

<u>Remarks</u>
a) The attribute c is the aforementioned 'underlying' attribute;
b) The attribute c is a restriction of the funtion TYPE (restricted to the domain δNg): $c = \mathrm{TYPE} \mid \delta Ng$.

Figure 2 shows how this type of generalization is usually drawn. The generalization δNg of δN_1 and δN_2 is characterized as
$Ng(\underline{a_1,\ldots,a_k},a_{k+1},\ldots,a_\ell,c)$
where $\{a_1,\ldots,a_k\}$ denotes the common key, a_j $(k+1 \leq j \leq \ell)$ denotes a common non-key attribute and c the underlying attribute.

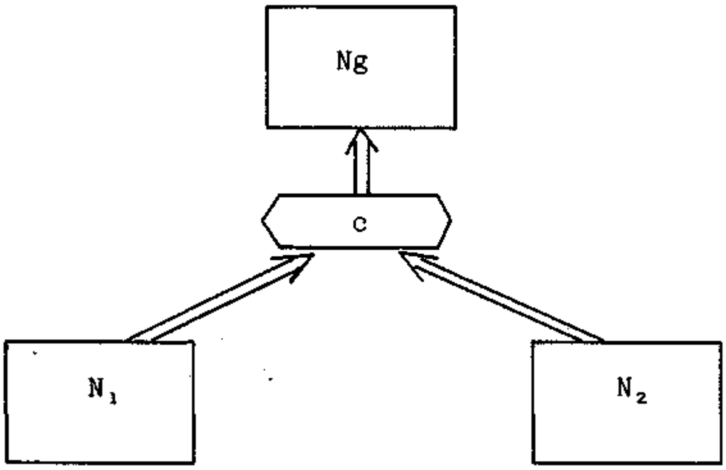

Figure 2

3.1.4 Third extension: subset hierarchy

Subset hierarchy is the second form of generalization[25]. We might call it non-exclusive generalization because the subsets are not necessarily disjunct. This also implies the absents of the underlying attribute (see figure 3).
Let δN_1 be an entityset and δN_2 a subset of δN_1 ($\delta N_2 \subseteq \delta N_1$).
If a_i is an attribute defined on δN_1 (a_i: $\delta N_1 \to V(a_i)$) then it is also defined on δN_2. The subset inherits all attributes of the superset; moreover, it may have further attributes for if $\delta N_2 \subset \delta N_1$ and a_i: $\delta \hat{N}_2 \to V(a_i)$ then a_i is not necessarily defined on δN_1.

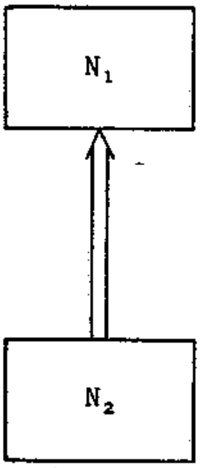

figure 3

The epistemological concepts obtained until now from the objectsentences have all the same mathematical shape:
$N(\underline{a_1,\ldots,a_k},a_{k+1},\ldots,a_n)$, primary entitysets as well as aggregates, generalized sets and supersets.
In other words we infer that given an Extended Entity Relationship (EER) Model as described above, there exists a function which maps the model into a collection of concepts of the form
$N(\underline{a_1,\ldots,a_k},a_{k+1},\ldots,a_n)$.

3.2 Modelling operations

The next step in the formal reconstruction of knowledge is dedicated to behaviour modelling.

The behaviour of the expert system as we view it is a result of an interpreter in action led by the prescriptions of the model. Thus a behaviour model hides only potential behaviour. Several attempts have been made to extend the original ER approach to include behaviour modelling[26], mostly strong related to Petri nets[27].

Also research is made to include notions like generalization[28] but difficulties with inheritance of properties keep us from defining a nice and clean method.

The source of the behaviour model is the set of operation sentences.

The operation sentences mention potential transitions of states, which we might classify into modules. These modules refer to certain entitysets subject to potential changes.

An entity is given a state by its attribute values. States are functions from entities into their attribute values.

Let s be a state, then $s: \delta N \rightarrow a_1[\delta N] \times \ldots \times a_n[\delta N]$

Limited states are defined to be functions from entities into values concerning a single attribute ($s^i : \delta N \rightarrow a_i[\delta N]$).

An extended state is a tuple of limited states: $s^* = (s_1, \ldots, s_n)$, where s_i is a limited state function from an arbitrary entityset δN into one of its attribute value sets.

States may change due to state transitions which we will call operations. A module maps extended states into extended states. Let M denote a module then $M(s^*) = s^{*\prime}$. In other words M refers to certain entitysets as being the input for potential transitions and also to certain entitysets as being target of the output of M[29].

Figure 4 shows a diagrammatic form of a module in relation to entitysets.

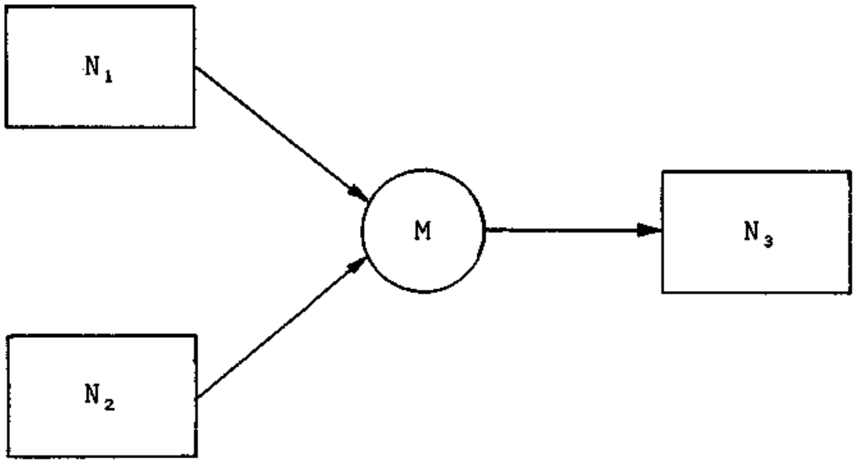

figure 4

3.3 Modelling events

An *event* happens when a module successfully executes, given the existence of relevant values (states of entities: the pre-condition) producing values (new states of entities: the post-condition). The event sentences mentioned in chapter 2 provide for the conditions required to transitions to actually take place. This includes potential events which may be interpreted as (heuristic) rules. So viewed, a rather classical modelling approach obtains new value in the area of knowledge engineering when used to build prototype knowledge based systems.[30]

Since we already developed the basis for a *database* (the EER-concepts) and a *modulebase*, it seems logical to define a *rulebase* where references to entitysets (conditions) and to modules (transitions) are combined into rules.
Work on this topic can be found in Bonczeck et al.[31], who use a network model, and Dahl[32], Dutta and Basu[33], who apply the principles of first order logic. The rulebase we are about to define differs from the usual notion of rules[34] because rules are no longer only if-then constructions, but may include algoritms.
A significant advantage of this approach is the distribution of dynamics over the model instead of putting it all together into one giant interpreter.

Let t denote a term, which is either a constant, a variable or a n-place function symbol. We substitute terms t_i for the attributes a_i in the EER-concept and consider each EER concept to be a *predicate* asserting some facts. An instantiation of the predicate evaluates to 'true' is it satisfies the facts, and 'false' otherwise.
Quantifiers are needed to characterize the introduced variables leading to *well-formed formulas* (wff's)[35] like $(\forall x)(\exists y)\ N(x,y)$ for example.
An *interpretation* of a wff is an assignment of values to the variables of the wff which might be either the input or the output of a module. Because wff's may always be transformed into Skolem-standard format[36] the pre- and postconditions of a module can be written as a conjunction of EER concepts with terms as arguments and all variables implicitly universally quantified.
A similar treatment concerns modules: substitute terms for limited states.
Let $\underline{M}$ be a characteristic function: $\underline{M}: M \rightarrow \{0,1\}$
$\underline{M}$ evaluates to 'true' if M is succesfully executed, otherwise $\underline{M}$ evaluates to 'false'. $\underline{M}$ is written as
$\underline{M}(t_1,\ldots,t_k,t_{k+1},\ldots,t_n)$ where $t_1,\ldots,t_k$ indicate the input of $\underline{M}$ and $t_{k+1},\ldots,t_n$ indicate the output.

We now can define a rule: A rule is the conjunction of one or more EER concepts and precisely one module of the type $\underline{M}$ with terms as arguments and all variables implicitly universally quantified.
For example $N_1\ (x, c, f(y)) \wedge N_2\ (x,y) \wedge \underline{M}\ (\underline{x},y,v) \wedge N_3\ (x,v)$ is a valid rule instantiated by a predicate form question like
$\exists x\ \exists y\ \exists v\ [N_1(x,20,f(y)) \wedge N_3(x,v)]$.
The aforementioned rulebase is defined to be the set of all rules.
The process of transformation from the level of informal representation to the level of formal representation described here, is a mapping from objectsentences into a database (using the EER-approach), a mapping from operation sentences into a modulebase (using behaviour modelling) and a mapping from eventsentences into a rulebase (using a first-order-logic technique).

4. CONCLUSIONS

The central issue of this paper is a formal treatment of an extension of the Entity-Relationship Approach in order to link up with first-order-logic oriented representation methods. We introduced a new image of the EER-approach assigning type-names to sets of entities. This gives us the opportunity to formally define the 'underlying' attribute in a generalization hierarchy.
We demonstrated how the EER-approach can be used as a modelbuilding-approach to design a prototype of a 'knowledge' base composed of three parts: a database, a module base, and a rulebase.
Due to the distribution of procedural 'knowledge' over the model a relatively simple interpreter may suffice to realize the required body of knowledge.

To the front of the EER-approach we plead for the use of several psychological eliciting methods since we are dealing with various kinds of knowledge at various levels.
Informal descriptions produced by these methods must endure a fase of linguistic analysis, in search of ambiguities, restricting the language according to suitable conventions, asking the expert for additional specifications, all in order to achieve a comprehensible unambiguous description. Finally we discerned three classes of sentences: object-sentences, operation sentences and event sentences. They constitute the basis for the extended Entity-Relationship Approach.

REFERENCES

Antonellis [1980]
V. De Antonellis et al. 'Extending the Entity-Relationship Approach to take into account historical aspects of systems', Procs., Int. Conf. on ERA 1980, P.P. Chen (ed.), North-Holland.

Antonellis [1985]
V. De Antonellis and B. Demo, 'Requirements collection and analysis' in 'Methodology and Tools for Data Base Design', S. Ceri (ed.), North-Holland, 1985.

Barr [1983]
A. Barr and E.A. Feigenbaum, 'The handbook of Artificial Intelligence', Volume I, II and III, Pitman Books Ltd., London (1983).

Batini [1984]
C. Batini and M. Lenzerini, 'A Methodology for Data Schema Integration in the Entity Relationship Model', IEEE Transactions on Software Engineering, vol. SE-10, no. 6, november 1984.

Bonczeck [1979]
R.H. Bonczeck and A.B. Whinston, 'The integration of network data base management and problem resolution', Information Systems, Vol. 4, 1979, pp. 143-154.

Bonczeck [1981a]
R.H. Bonczeck et al., 'A Generalized Decision Support System using predicate calculus and Network Data Base Management', Operations Research, Vol. 29, No. 2, 1981, pp. 263-281.

Bonczeck [1981b]
R.H. Bonczeck et al., 'Foundations of Decision Support Systems', Academic Press, London, 1981.

Bonczeck [1982]
R.H. Bonczeck et al., 'The evolution from MIS to DSS: Extension of Data Management to Model Management', in 'Decision Support Systems', (M. Ginzberg et al. eds.), North Holland, 1982.

Bosman [1985]
A. Bosman and H.G. Sol, 'Knowledge Representation and Information System Design', in 'Knowledge Representation for Decision Support Systems', L.B. Methlie and R.H. Sprague (eds.), North-Holland, 1985.

Brodie [1984]
M.L. Brodie et al. 'On Conceptual Modelling', Springer-Verlag, New York, 1984.

Chang [1973]
C. Chang and R.C. Lee, 'Symbolic Logic and Mechanical Theorem Proving', Academic Press, London, 1973.

Chen [1976]
P.P. Chen, The Entity-Relationship Model - Toward a Unified View of Data', ACM-TOD, Vol. 1, No. 1, 1976, pp. 9-36.

Clancey [1983]
W.J. Clancey, 'The Epistemology of a Rule-based Expert System - a Framework for Explanation', Artificial Intelligence 20, 1983, pp. 215-251.

Codd [1979]
E.F. Codd, 'Extending the Database Relational Model to Capture More Meaning' ACM-TOD, Vol. 4, No. 4, 1979, pp. 397-434.

Dahl [1982]
V. Dahl, 'On Database Systems Development Through Logic', ACM-TOD, Vol. 7, No. 1, 1982, pp. 102-123.

Dutta [1985]
A. Dutta and A. Basu, 'Representation and Manipulation of Modelling Knowledge in Decision Support Systems', in 'Knowledge Representation for Decision Support Systems', L.B. Methlie and R.H. Sprague (eds.), North-Holland, 1985.

Gammack [1985]
J.G. Gammack and R.M. Young, 'Psychological Techniques for Eliciting Expert Knowledge', in 'Research and development in expert systems', M.A. Bramer (ed.), Cambridge University Press, 1985.

Hayes-Roth [1985]
F. Hayes-Roth, 'Rule-Based Systems', Communications of the ACM, Vol. 28, No. 9, 1985.

Kleefstra [1980]
W.J. Kleefstra, 'The concept of Datamodel in Data Base Systems', (Ph.D. Thesis), Krips Repro Meppel, Holland, 1980.

Mourelatos [1978]
A.P.D. Mourelatos, 'Events, Processes and States', Linguistics and Philosophy, 2, 1978, pp. 415-434.

Newell [1982]
A. Newell, 'The Knowledge Level', Artificial Intelligence 18, 1982, pp. 87-127.

Peterson [1977]
J.L. Peterson, 'Petri Nets', Computing Surveys, Vol. 9, No. 3, 1977.

Roland [1983]
C. Roland, 'Database Dynamics', Data Base, Spring 1983.

Sakai [1983]
H. Sakai, 'A Method for Entity-Relationship behaviour modeling', in Procs. Int. Conf. on ERA, 1983, C.G. Davis et al.(eds.), North-Holland.

Schefe [1982]
P. Schefe, 'Some Fundamental Issues in Knowledge Representation', Informatik-Fachberichte No. 58, GWAI-82, Springer Verlag, 1982.

Schiffner [1979]
G. Schiffner and P. Scheuermann, 'Multiple Views and Abstractions with an Extended Entity-Relationship Model', Computer Languages, Vol. 4, 1979, pp. 139-154.

Smith [1977a]
J.M. Smith and D.C.P. Smith, 'Database Abstractions: Aggregation', Communications of the ACM, Vopl. 20, No. 6, 1977, pp. 405-413.

Smith [1977b]
J.M. Smith and D.C.P. Smith, 'Database Abstractions: Aggregation and Generalization', ACM-TOD, Vol. 2, No. 2, 1977, pp. 105-133.

Solvberg [1984]
A. Solvberg and C.H. Kung, 'On Structural and Behavioral Modelling of Reality', Techn. Report no. 25/84, Univ. of Trondheim, Div. of Comp. Science, 1984.

Sowa [1984]
J.F. Sowa, 'Conceptual Structures: Information Processing in Mind and Machine', Addison-Wesley, London, 1984.

Tsichritzis [1982]
D.C. Tsichritzis and F.H. Lochovsky, 'Data Models', Prentice-Hall, New Jersey, 1982.

Webre [1983]
N.W. Webre, 'An Extended Entity-Relationship Model and its use on a Defense Project', in 'Entity-Relationship Approach to Information Modeling and Analysis', P.P. Chen (ed.), North-Holland, 1983.

NOTES

1. Chen [1976]
2. Schefe [1982, p. 44]
3. Schefe [1982, p. 45]
4. Kleefstra [1980, p. 24]
   Sowa [1984, p. 11, p. 70]
5. Newell [1982, p. 98]
   The system (called agent) at the knowledge level is composed of a set of actions, a set of goals and a body. The medium is knowledge. The system acts according to the principle of rationality.
6. Bosman [1985, p. 83]
7. Gammack [1985, p. 106]
8. Antonellis [1985, p. 15]; Batini [1984]
9. Antonellis [1985, p. 15]
10. Mourelatos [1978, p. 423]
11. It is obvious that this convention implies restrictions to the transformation of the informal descriptions of the expert's knowledge. For instance, heuristic rules might be considered as conditional state transition descriptions, but strategic knowledge is likely to disappear in this convention. Nevertheless our aim is to provide for a tool to build (incomplete) prototypes.
12. Roland [1983]
13. Tsichritzis [1982, p. 223]
14. Antonellis [1985, p. 15]
15. Antonellis [1985, p. 21]
16. Chen [1976]
17. Sowa [1984, p. 79]
18. Sakai [1983, p. 112]
19. Smith [1977a]; Smith [1977b]; Schiffner [1979}
20. Codd [1979, p. 418]
21. Webre [1983, p. 189]
22. Sakai [1983, p. 113]
23. Smith [1977b]; Schiffner[1979]; Codd [1979]
24. Smith [1977b, p. 111]
25. Schiffner [1979, p. 142]
26. Solvberg [1984]; Antonellis [1980]; Sakai [1983]
27. Peterson [1977]
28. Brodie [1984]: various articles dedicated to this topic, especially that from A. Borgida et al. p. 87.
29. We deliberately avoid going into details concerning the partial character of M. In our view M is an algoritm that, once executed and given desired input, produces some output.
30. Clancey [1983]; Hayes-Roth [1985]
31. Bonczeck [1979]; Bonczeck [1981a]; Bonczeck [1981b]; Bonczeck [1982, p. 61]
32. Dahl [1982]
33. Dutta [1985, p. 93]
34. Barr [1983, part I, p. 190]
35. Chang [1973, p. 29]
36. Chang [1973, p. 47]

Serie Research Memoranda (1)

1981-1	E. Vogelvang	A quarterly econometric model for the Price Formation of Coffee on the World Market
1981-2	H.P. Smit	Demand and Supply of Natural Rubber, Part I
1981-3	R. Vos	The political Economy of the Republic of Korea; A proposal for a model framework of an open economy in the ESCAP-region, with emphasis on the Role of the State
1981-4	F.C. Palm	Structural Econometric Modeling and Time Series Analysis - Towards an Integrated Approach
1981-5	P. Nijkamp in co-op. with H.v. Handenhoven and R. Janssen	Urban Impact Analysis in a Spatial Context: Methodology and Case Study
1981-6	R. Ruben	Primaire exporten en ekonomiese ontwikkeling
1981-7	D.A. Kodde	Het genereren en evalueren van voorspellingen van omzet en netto winst: een toegepast kwantitatieve benadering
1981-8	B. Out	Financiële vraagstukken onder onzekerheid
1981-9	P.van Dijck and H. Verbruggen	A Constant-Market-Shares Analysis of ASEAN Manufactured Exports to the European Community
1981-10	P. Nijkamp, H.de Graaff and E. Sigar	A Multidimensional Analysis of Regional Infrastructure and Economic Development
1981-11	P. Nijkamp	International Conflict Analysis
1981-12	A.J. Mathot	L'Utilisation du Crédit lors de l'Achat d'une Voiture
1981-13	S.P. van Duin en P.A. Cornelis	Onderzoek naar levensomstandigheden en opvattingen over arbeid bij mensen zonder werk, deel I
1981-14	W. van Lierop and P. Nijkamp	Disaggregate Models of Choice in a Spatial Context
1981-15	Hidde P. Smit	The World Vehicle Market
1981-16	F.C. Palm	Structural Econometric Modeling and Time Series Analysis: An Integrated Approach
1981-17	F.C. Palm and Th.E. Nijman	Linear Regression Using Both Temporally Aggregated and Temporally Disaggregated Data
1981-18	F.C. Palm and J.M. Sneek	Some econometric Applications of the exact Distribution of the ratio of Two Quadratic Forms in Normal Variates