COMPARISON OF FOUR RELATIONAL DATA MODEL SPECIFICATIONS

Winfried Lamersdorf

Universität Hamburg
Fachbereich Informatik
Schlüterstrasse 70
D-2000 Hamburg 13
West-Germany

Abstract:

This paper reviews four different approaches to specifying the relational data model which are written at about the same time:

1. the 'Definition and Formalization of the Relational Data Model' as provided as part of the final ANSI/X3/SPARC/DBSSG Relational Database Task Group Report,

2. the 'Relational Database Model Specifications' as submitted to the National Bureau of Standards by the Computer Corporation of America as a draft report in October 1981,

3. the 'Formal Definition of the Relational Model' as specified by C. Date of IBM, San Jose, and

4. the specification of the relational data model which is expressed in terms of the 'Positional Set Notation' as given in the 'Abstract Database Model' project of the National Bureau of Standards in October 1981.

The paper points out some of the differences between the four approaches to specifying the RDM semantics.

This work was done, in part, while in residence at the Data Management and Programming Languages Division, Center for Programming Science and Technology, Institute for Computer Sciences and Technology, National Bureau of Standards, Washington, D.C. 20234

TABLE OF CONTENTS

1. INTRODUCTION

During the ten years since its first publication, the Relational Data Model (RDM) has attracted increasing interest and is now widely accepted. Similar to recent developments in programming languages, definitional issues shifted from syntactical aspects as realized in particular relational database management systems (RDBMSs) to the semantics of the RDM. There are, however, different perceptions of the RDM as well as alternative ways of expressing its semantics.

The purpose of this paper is to review four semantic specifications of the RDM which have been written recently (chapter 2). It also compares the four specifications with respect to the specification methods used and their respective perceptions of the RDM (chapter 3).

2. FOUR RELATIONAL DATA MODEL SPECIFICATIONS

2.1 THE 'RTG' SPECIFICATION

The definition and formalization of the relational data model [ScLa82] which is part of the ANSI/X3/SPARC/DBSSG Relational Database Task Group's (RTG) final report [BrSc82] is divided into two different parts: an informal description of the basic concepts of the RDM, and a formal semantic specification of its structural and operational constituents.

2.1.1 The Informal Definition of the RDM

In chapter 2 of the final RTG report [BrSc82], an informal definition is given for the basic concepts underlying the RDM. This definition is written in a rather abstract way. It attempts to define a core of concepts which could be viewed as being a common basis for most of the different RDM implementations.

In the informal part, the RDM is defined in terms of its

1. structuring mechanism (domains, attributes, tuples, relations, databases),
2. constraint mechanism (with relation keys as the most important example), and
3. operational means.

Operations on relations are divided into queries and alteration operations. As queries are concerned, a balanced characterization is given for both relational algebra and relational calculus expressions. The basic algebra operations are outlined, and the the basic expressions of the relational calculus are characterized. In addition, the semantics of the three fundamental relation alteration operations are defined, and alternative ways of handling special cases are indicated.

All through this part of the RDM specification, several examples are given to demonstrate structure definitions and operation applications using the syntax and semantics of various database management systems.

The main idea of the informal part of the RDM specification is to give a general introduction to the most important relational terms and to define a 'core' of relational concepts which could be agreed upon by the relational database community.

2.1.2 The Formal Definition of the RDM

In the second part of the RDM definition [ScLa82] which is included in the appendix of the final RTG report [BrSc82], the semantics of the RDM 'core' is defined formally and in detail. As an appropriate semantic specification method the 'Vienna Development Method' (VDM) is used. VDM is described in detail elsewhere [BjJo78].

In the first part of the formal RDM specification ('Semantic Domains'), an abstract description is given for the structural aspects of the data model. The 'Static Consistency Constraints' specify additional structural constraints which apply to any 'well-formed' relational database.

In the second part ('Syntactic Domains'), an abstract syntax description is given for several important operations of the RDM. 'Dynamic Consistency Constraints' are specified to define further restrictions refering to possible operands of any 'well-applied' relational operator.

Finally, in the 'Elaboration Functions'-part of the specification, the semantics of the operations is defined by specifying abstract ('semantic meaning') functions which denote the meaning of these operations according to the denotational semantics approach.

2.2 THE 'CCA' SPECIFICATION

The RDM specification of the Computer Corporation of America (CCA) [Piro82] is expressed in a partly informal but mainly semi-formal way. After a short introduction to the semi-formal notation used in this specification, the basic relational terms (domains, relations, constraints, databases) are defined both informally and using the semi-formal notation. The specification emphazises the notion of domains for relation definitions. Relation keys are not considered as being essential (they are treated as one out of several possible constraints). A hint on the common way to view relations as 'tables of rows' concludes the descripton of the structural aspects of the data model.

The main part of the 'CCA' RDM specification consists of a detailed definition of all the major relational algebra operations. To achieve the goal of a general closure property for relation algebra operations the specification heavily relies on a renaming operation for attributes. After the description of an operation 'rename', all basic algebraic operations are specified semantically. Included is a general Cartesian product operation. Different kinds of 'join' and 'division' operations are classified. Several possible extensions to the relational algebra operations are mentioned explicitly.

After the semantic specifications, a possible ('concrete') syntax definition is given for algebraic expressions which are written in a purely applicative way.

The CCA report specifies two different relational calculi as possible ways for expressing non-algebraic queries: a tuple oriented relational calculus (TRC) and a domain oriented relational calculus (DRC). Both calculi rely on a relation viewed as a ('membership') predicate which is defined for all possible tuple values of a relation type. It yields 'true' for every tuple value which is part of an relation's actual value and 'false' otherwise.

In the definition of the calculi, the meaning of variables, the kinds of queries, atomic formulas, and formulas are specified in detail. Possible evaluation strategies for queries are not considered. Finally, the ('concrete') syntax of both kinds of calculi is summarized by use of context-free BNF-grammars.

The CCA report mentions explicitly mechanisms to invoke external functions as part of the RDM.

In its last section that describes the operations, the CCA report specifies the basic relation alteration operations by giving both an informal and a semi-formal description of their respective semantics.

The report also mentions possibilities of writing explicit assertions in the RDM. It furthermore specifies two possible tuple-at-a-time interfaces to an environment of a relational database system.

2.3 THE 'IBM' SPECIFICATION

Similar to the 'RTG' definition, the definition of the RDM given by C. Date of IBM, San Jose [Date82], is divided into three parts: a definition of the set of objects ('relational database'), the set of operations ('relational operations'), and the set of general integrity rules ('relational rules'). Although this definitiion of the RDM is called 'formal', major parts of the RDM semantics are provided in informal English.

The structural definition of the relational objects covers all the basic concepts of domains, relations (as variables), tuples, attributes, keys etc. It is summarized by means of a set of production rules which are explained by detailed comments.

The specification of the queries is restricted to a minimal set of relational algebra operations. Some additional operators are mentioned (informally) in a separate chapter. No 'rename' operation is specified explicitly. However, attribute renaming is implicitly used in the description of some other operations (see 5.3.5).

According to the minimal approach, relation altering operations are purely defined in terms of a general relation assignment operation and an appropriate use of of the 'union' and 'difference' operators.

Finally, two classes of relational integrity constraints are specified: an 'entity' rule for intra-relational integrity constraints, and an inter-relational referential integrity rule.

2.4 THE 'NBS' SPECIFICATION

In the 'Abstract Database Models' project of the 'Institute for Computer Science and Technology' in the National Bureau of Standards (NBS) [NBS81], the 'Positional

Set Notation' (PSN) [Hard82] is used as a tool to define the semantics of data models formally. Such specifications can be processed by a software system, the 'data model processor' (DMP) which emulates the behaviour of the specified data model [KoHS82]. (This fact, however, is not considered any further here.)

So, in the DMP framework, PSN is used as a specification method to express all structural and operational aspects of a data model in a formal way. Thus, PSN provides a meta- or specification language and can be compared to the other ways of specifying data model semantics formally.

The first task ('human role') to be performed for the DMP is the specification of a new data model and is called the 'data model definition'. This specification begins with an introduction of all new 'concepts' involved. In case of the RDM specification as demonstrated in [NBS81] and [KoHS82], the following concepts have to be defined: (explicit and implicit) domains, relations (types), relation occurrences, and some kind of access control information for the different relations.

In the second part of this specification, the newly introduced concepts are defined in terms of 'positional sets' (P-sets) [Hard82]. First, the range variables used in the P-set definitions are listed together with the P-sets they are bound to. Then, the P-sets are defined based on PSP 'template' descriptions. The 'templates' specify the structural aspects of the introduced concepts as well as some additional structural constraints. The constraints are expressed in terms of general 'where clauses' in the template definitions. These clauses are similar to the predicates in calculus-oriented relational query languages (e.g. PASCAL/R, INGRES, SYSTEM R).

In the last and main part ot the RDM definition, the data model's operational aspects are specified. As these specifications are to be processed by the PSP, they consist, in essence, of (machine executable) PSP commands. In some cases, additional specifications or control structures have to be given by simple C-code statements.

The ('primitive') relational operations specified in [NBS81] are

1. a complex 'create' operation which builds up new relations (occurrences and possible types) according to conditions specified as operation parameters (the relational 'select', 'project', and 'join' operations can be expressed in terms of the PSP 'create' operation.),

2. a 'remove' operation for relations (types as well as occurences),

3. set operations for relations (union, intersection, and symmetric difference), provided the respective relation (type) definitions are equal,

4. formatting and print operations to output relation (occurrence) values,

5. relation alteration operations (insert, delete, and replace) at a tuple-at-a-time level (tuples to be deleted or replaced are identified by a general 'condition' parameter), and

6. some auxiliary operations which return, e.g., the set of all relation names, the attribute names of a given relation, the key attribute names only, the domain-name which is associated with a given relation attribute, or membership operations which test whether a certain value is contained in the set of values of a given domain.

3. DIFFERENCES OF THE FOUR SPECIFICATIONS

3.1 DIFFERENCES OF PURPOSE

The RTG report tries to specify a 'minimal' relational database model which could (and should !) be the core of any RDBMS. The definition may be thought of as a global intersection between the sets of functions offered by current RDBMSs.

The CCA report tries to specify a more comprehensive overview over many different essential components of the RDM as well as possible variations. This may be thought of as a global union over all possible features of RDBMSs. The 'IBM' definition aims at a specification of the basic relational concepts and concentrates on a small (but relationally complete) set of operations. Possible extensions which may be part of actual relational systems are mentioned but not specified in detail.

Contrary to that, the main purpose of the 'NBS' specification of the RDM is to demonstrate the capability of the PSP to specify (and then emulate) the semantics of major database models. So, the choice of the structural and operational aspects as well as the imposed constraints are not necessarily meant to be 'standard' for the RDM in a general

sense. The PSP, however, provides a set of tools which are powerful enough to define the semantics of any 'standard' RDM definition operationally.

3.2 DIFFERENCES IN THE SPECIFICATION METHODS

The RTG report relies, in its first part, on an informal, natural language introduction to the data model using examples refering to several different systems. In its second part, it gives a more detailed specification of the semantics using a completely formal semantic specification method. This leads to a single comprehensive and consistent formal model for the semantics of all aspects of the RDM.

The CCA report is a combination an informal introduction to the basic RDM components and a semi-formal specification of the RDMs most important parts. The semi-formal descriptions for the different components are modeled seperately and are only loosely connected. The authors of the CCA report believe that this combination is the best compromise between precision and understandability of the semantic specification.

In the 'IBM' specification, the relational database structures are summarized by means of a set of (BNF-like) production rules for the abstract syntax of all RDM objects. The specification is called 'formal', although major semantic explanations as well as additional integrity constraints are given informally in natural language comments to the production rules. The abstract structure of the operations is summarized in a similar way. The operations' semantics is defined in terms of semi-formal comments to the production rules of the operations, based on concepts specified in either the structural or the operatonal production rules.

The 'NBS' specification of the RDM uses PSN and the PSP commands as the meta-language to descibe the structural and oparational components of the data model together with the their constraints. As the specification is intended to be machine executable, it has to be completely formal in that sense. Only in cases where PSP commands are not powerful enough, the specification is augmented by some lines of (executable) C-code. This specification emphasizes machine executability of the definition rather than comprehensiveness.

3.3 DIFFERENCES IN THE SPECIFIED DATA MODELS

Amongst the four specifications there are only slight differences in the perception of the RDM. The differences are related to the following concepts:

3.3.1 Domains

The CCA report emphasizes the role of domains. They provide values as well as the operations to be applied to them. The 'CCA' RDM specification is more restrictive than the 'RTG' specification in the sense that it allows operations on values on the same domain only. Furthermore, it requires all sets of domain values to be disjoint. The 'RTG' definition just mentions the role of domains in connection with type compatibility of tuples and relations. The 'IBM' RDM definition introduces an explicit 'ordering-indicator' for each set of domain values, and it regards 'null' as a possible domain value. The 'NBS' specification defines two distinct, alternative concepts of an explicit and an implicit domain definition. A list of attributes is attached to each domain definition describing explicitly its value-set, length, set of compatible domains, constraints, etc.

3.3.2 Tuples

Tuples are mentioned as a concept of their own in the RTG report. In the three other specifications they are only part of the concept of a relation's value.

3.3.3 Relations

The CCA report emphasizes the table view of a relation; it also relies on a set-oriented view for the specification of the relational algebra operations. A predicate-oriented view of the relation value is mentioned in the specification of the relational calculi only.

Throughout the RTG report a relation is described in both ways, as a set and as a predicate. The RTG report purposely does not give preference to either one of these two perceptions.

The 'IBM' specification describes relation values as tuple sets. It makes a distinction between the conceptes of a 'real' relation (variable), a named 'virtual' relation, and an (unnamed) relation expression. The 'NBS' specification favours the set-oriented view of a relation (occcurence) value. The 'NBS' specification is the only one that considers the definition of a relation type independently of possible (value) occurences of that type.

3.3.4 Integrity Constraints

In the CCA report, keys are not mentioned explicitly as being part of the concept of a relation. The RTG report describes the key concept as the most important example for a relational constraint mechanism. The 'NBS' specification defines some constraints for relation occurrences as well as

for relation types. The key constraint is one of them; but in [NBS81] its specification is not given in complete detail. The IBM paper mentions the concepts of a primary, an alternate, and a candidate key for a relation variable. The respective constraints, however, are explained only informally.

In the RTG report, inter-relational constraints are mentiond, but not specified in detail. The 'IBM' definition gives a semi-formal specification for a referential integrity rule.

3.3.5 Operations

The 'CCA' specification of the algebra operations is based on a renaming operation for attributes that guarantees the closure property of the relational algebra. The RTG report specifies a renaming operation in its formal part as well. However, in general, it leaves the question open, whether to solve attribute naming problems by renaming or by additional constraints on the operations (as some relational database systems do). The 'NBS' specification allows renaming of attributes indirectly as an option of the more general 'create' command for relations. The 'IBM' definition does not specify an explicit rename operation. However, default renaming rules are applied in the semantic description of the 'union' and 'product' operations. The relation assignment operation requires an explicit specification of the correspondence between the two sets of attributes involved.

Union-compatibility is specified formally in the 'RTG' definition and in a semi-formal, precise, but less restrictive way in the 'CCA' and in the 'IBM' definitions.

The CCA report mentions explicitly many possible relational algebra operations which are part of some relational systems and which are missing in others (e.g. a general Cartesian product, different kinds of join and division operations, possible extensions to other operations). Both, the 'RTG' and the 'NBS' definition specify only a basic set of the most common relational algebra operations. The IBM paper specifies a basic set of primitive algebra operations, and then refers to possible extensions in a separate chapter.

The RTG report only mentions the basic constituents of relation calculus expressions. It does not define different kinds of relational calculi. The CCA report - according to its emphasis on domains - treats a domain oriented relation calculus separately besides the usual tuple oriented one. The 'IBM' specification does not consider the relation calculus at all. In the 'NBS' definition, the relation algebra operations 'select', 'project', and 'join' are specified by means of a more calculus oriented, general relation 'create' operation.

(This approach is straightforward because of the 'create' operation on the PSP.)

In addition to the semantics of the RDM, the CCA report provides a ('concrete') syntax for the relational algebra as well as for both relational calculi. Neither the 'RTG' nor the 'NBS' nor the 'IBM' specification define any kind of (concrete) syntactical aspect of the data model.

The relation alteration operations are defined in a rather similar way in the RTG and the CCA reports. The three main operations are called 'insert', 'delete', and 'replace' in the RTG report, and 'store', 'delete', and 'modify' in the CCA paper. In both cases, the relation alteration operations are specified in a set oriented way, based on relations and relation values in general (as opposed to tuple oriented only). The tuple identifiers needed in order to identify the tuples to be altered by the 'replace'/'modify' operation are given as a subset of the tuple attributes in the CCA report. They are not further specified in the RTG report. In the 'NBS' specification, all three relation alteration operations are defined similarly. 'Insert', however, is defined on a tuple-at-a-time base. The 'delete' and the 'replace' operations identify the tuples to be deleted/replaced by a general condition rather than by providing the set of tuples explicitly or implicitly through their keys. The IBM paper specifies only a basic relation assignment operator. The semantics of further relation alteration operations has to be composed out of this operation and the respective relational expressions.

Only three reports mention explicitly sets of tuples coming from outside the database as relation-valued operands in relation alteration operations (the 'external' relations in the RTG report, the 'literal' relations in the CCA paper, and the 'relation-literal' in the IBM report.)

The RTG, IBM, and the NBS reports do not mention assertion facilities, external interfaces, or function invocation mechanisms as part of the RDM. Only the 'NBS' specification defines some additional output, format, and auxiliary operations.

3.3.6 Access Control

The 'NBS' definition specifies access control information related to single relations, attributes, and operations as an additional structural aspect of the data model. None of the other RDM specifications mentions access control explicitly. However, relation views are considered in the RTG report and in the IBM paper ('virtual relations').

4. CONCLUDING REMARKS

All four specifications of the RDM have been developed at about the same time. They all aim at a precise semantic definition of the RDM's main structural and functional aspects. The describing 'meta-language' is in all cases based on some kind of formal notation.

The specification is completely formal in case of the 'RTG' and the 'NBS' definitions. The other two are based on a mixture of some formal specifications and informal explanations. Comparatively, the formal notation used in the CCA report is more concise and precise than that of the IBM paper.

The main goal of the CCA and of the IBM papers is to give a precise and understandable semantic definition of the RDM by introducing a formal meta-notation into a - basically - natural language description. The 'NBS' specification, on the other hand, is less understandable but completely formal in the sense that it can be interpreted automatically. The 'RTG' specification, finally, uses a formal notation to achieve unambiguity while still maintaining readability.

There are only slight differences in the object systems specified: the CCA report defines an extensive set of many different relational algebra operations, whereas the other three specifications rather tend to concentrate on a basic (or even minimal) choice of 'core' relational operations.

5. REFERENCES

[BjJo78] D. Bjørner, C.B. Jones (Eds.) : 'The Vienna Development Method: The Meta Language', Lecture Notes in Computer Science, No. 61, Springer Verlag, Berlin Heidelberg New York, 1978.

[BrSc82] M.L. Brodie, J.W. Schmidt (Eds.): 'Final Report of the Relational Database Task Group', ANSI/X3/SPARC/DBSSG, ACM SIGMOD RECORD, Vol. 12, No. 4, July 1982.

[Date82] C.J. Date: 'A Formal Definition of the Relational Model', TR 03.169, IBM General Product Division, San Jose, California, October 1981 - also available in: ACM SIGMOD RECORD, Vol. 13, No. 1, September 1982.

[Hard82] W.T. Hardgrave: 'Positional Set Notation', Advances in Database Management, Vol. 2, Heyden and Son, New York, to appear 1982.

[KoHS82] M. Koll, W.T. Hardgrave, S. Salazar: 'Data Model Processing', Proc. National Computer Conference, Houston, Texas, June 1982.

[NBS81] National Bureau of Standards, Institute for Computer Science and Technology, Center for Programming Science and Technology, Data Management and Programming Languages Devision: 'Abstract Database Models Competency Project Peer Review', Gaithersburg, Maryland, October 1981.

[Piro82] Computer Corporation of America: 'Relational Database Model Specifications', Draft Report submitted to the National Bureau of Standards Washington/DC, Cambridge, Massachusetts, October 1981 - a revised version of major parts of this paper is published as: A. Pirotte: 'A Precise Definition of Basic Relational Notions and of the Relational Algebra', ACM SIGMOD RECORD, Vol. 13, No. 1, September 1982.

[ScLa82] J.W. Schmidt, W. Lamersdorf: 'Relational Datamodel: A Definition and its Formalization', Bericht Nr. 88, Fachbereich Informatik, Universität Hamburg, West Germany, March 1982 - also available in [BrSc82].