VALIDATION, VERIFICATION, AND INTEGRITY IN KNOWLEDGE AND DATA BASE SYSTEMS: FUTURE DIRECTIONS

Frans Coenen
Department of Computer Science,
University of Liverpool, UK
frans@csc.liv.ac.uk

Barry Eaglestone
Department of Information Studies,
University of Sheffield,UK
B.Eaglestone@shef.ac.uk

Mick Ridley
School of Computing and Mathematics,
University of Bradford, UK
M.J.Ridley@scm.brad.ac.uk

Abstract

In this paper we consider possible future directions for rule base V&V on the presumption that many techniques developed by the rule base V&V community have come out of the "research domain" into what might be described as "the main stream". In particular we consider opportunities for possible "cross-overs" between database integrity checking techniques and a number of rule base V&V strands. Collectively we refer to these techniques as VVI (Validation Verification and Integrity) techniques. We commence by giving an overview of integrity issues, then go on to consider overlaps between database and rule base systems and finish with a discussion of "future directions" in VVI.
KEYWORDS: VVI (Validation, Verification and Integrity)

1 INTRODUCTION

The database community has been concerned with the issue of data integrity almost since the conception of database systems. Codd's 1970 paper [10] includes a section on "Consistency" that covers this ground (and data integrity occurs in the article's key words and phrases) although it is perhaps not until Codd's 1979 paper [11] that integrity rules are explicitly formulated as *entity integrity* and *referential integrity* in a way that will be familiar from Codd's 12 rules [12]. There is also now a considerable history of attempts to formalise this area from Nicolas [31] onwards. Consequently work on data base integrity issues considerably predates the work on rule base V&V.

It can be argued that the traditional view of rule base system V&V was established in the late 1980s/early 1990s and focussed on the static and dynamic inspection of rule bases to identify anomalies and symptoms of errors. Examples of this work include: Preece and Shinghal's COVER tool which checked KBSs written in a generic rule language ([33]), Suwas's rule checking program developed as part of the Work on MYCIN ([37]), Cragen and Steudel's ECS decision table based KBS V&V program ([14]), and Ginsberg's KB-reducer ([20]) (there are many others). Although software verification and validation has always been perceived as being unglamorous, this initial work was prompted by a clear requirement for rule base V&V tools as the KBS/expert system concept moved from a research platform to a commercial one. To a large extent this work is now generally accepted by practitioners and can be described as "main stream" technology, rather than research technology.

Current work on rule base V&V (much of which can be traced back to the earlier, now well established, work noted above) is concerned with the automated testing of systems and semi-automated revision of rule bases (KRUSTWORKS — [15]), the testing of task specific properties of rule bases ([22]), V&V at the formal specification level of KBS development (KIV — [19]), quality assurance ([39]) and the application of rule bases V&V tools to wider domains such as the WWW ([35]).

On the face of it there is very little common ground between the integrity considerations associated with databases and the V&V of rule bases. Integrity is concerned with the ongoing validity of dynamic systems, i.e. a constantly evolving database; while validation and verification are respectively concerned with the truthfulness of a system with respect to its problem domain and the correctness of its construction, i.e. whether a rule base is logically or structurally flawed. Integrity is not a concept with which KBS are generally concerned — possibly because such systems tend not to be dynamic. Conversely the terms V&V are rarely used in the database literature ([18]). (Of course in the broader sense, validation and verification forms an important part of the traditional software engineering of the Data Base Management Systems (DBMS) as it also does in the development of the inference engine component of expert systems.)

There has always been some acknowledgement that there is a kinship between the relational model of databases and rule bases systems in that both can be expressed in clausal form logic, however in the context of VVI (Validation, Verification and Integrity) the two communities have rarely come together. In more recent years, however, the establishment of alternative database models has begun to alter this situation such that a significant overlap between the two technologies can now be identified.

In this paper we firstly (Section 2) give an overview of the integrity issues that concern the database community (it is assumed that the intended readership of this paper does not require an overview of the concerns associated with KBS V&V). We then examine this growing overlap between the two genres in Section 3. Having established that significant overlap exists we then go on in Section 4 to identify opportunities (future directions) for the application of "cross-over" technologies — the aim being to widen the scope of rule base V&V to encompass VVI. Some conclusions are drawn in Section 5.

2 DATABASE INTEGRITY

Although, as noted above, integrity may have been a central concern to the database community and of database theory for many years, it also true to say that it has been poorly implemented in many database systems. Codd made the point:

> "It is ,however, vitally important to remember that the relational model includes three major parts: the structural part, the manipulative part and the integrity part ..." [12]

in the context of questioning the relational credentials of many systems. Similarly Date [16] in the mid 80s, in a standard introduction to database systems, said:

> "..no system, SQL-based or otherwise, currently provides very much in the way of integrity support."

The situation has improved since then and in this section we outline the main issues from a theoretical position and note that the techniques of implementing these vary considerably from system to system.

The integrity part of the relational model consists of two general integrity rules:

Entity Integrity: Enforcing entity integrity ensures that there is a unique name or identifier in each row in a table to distinguish the object being represented from any other object.

Referential Integrity: Referential integrity is concerned with maintaining cross references from object to object in a database. Much of the power of relational databases derives from their ability to connect the items of data held in separate tables on common values. Enforcing referential integrity means that those values which are keys in tables are kept consistent.

Although the above refers to the relational model the concepts of entity and referential integrity can be extended to other models.

2.1 What are we maintaining?

The constraint support from most RDBMSs will stop changes to an existing database that violate the rules, which may be held in a number of ways, but do not check the entire database so a previous consistency is assumed. This may be significant since databases can be set up with bulk loading utilities that do not obey integrity rules.

Without an extensible type mechanism many constraints that are really (or could be) domain constraints may be represented by referential integrity constraints. For example if we want a data item to be a day of the week it may only be possible to achieve this by checking that an instance of this data item occurs in a table that simply contains "Monday","Tuesday", etc.

2.2 How are we maintaining it?

In RDBMSs integrity constraints will be enforced in a number of ways. Entity integrity is generally better supported than referential integrity and is usually provided via a standard SQL command that creates a table and declares that a column or columns are the Primary key and must not be null. This is a case where support has improved; in the past for example the Ingres RDBMS provided this support not by specifying a primary key, and hence uniqueness, but by making the table's storage structure one that guaranteed uniqueness. This meant that there was no guarantee that the storage structure could not be changed in the future so that the constraint could then be broken.

Referential integrity is commonly implemented by database rules and procedures. These may take a form like the ECA format discussed in section 3.3.2 (active databases). This type of mechanism allows changes to be cascaded through the database so that an update to one table is propagated to all references to that datum throughout the database. Conversely it can also be used to block changes so that data may not be removed from a table if there are references to it in other tables.

Finally, simple domain constraints may be supported by an SQL "create integrity" statement, and other domain constraints may be supported by the type system of the DBMS itself.

3 THE VARYING RELATIONSHIPS BETWEEN DATABASE AND RULE BASE SYSTEMS

In this section we review the varying relationships between database and rule base systems. We commence by firstly considering the architecture of both; and then examining the interconnection between rule base systems and different DB models commencing with the relational model and then going on to consider the deductive, active, object-oriented, and constraint models which have been proposed, with varying degrees of success, in more recent years. The objective of this section is to establish the nature of the overlap between the database and expert system technologies.

3.1 Architecture

Initially, the most obvious mechanism for high lighting the similarities between database and expert systems is to consider the components of such systems. A diagram is presented in Figure 1. Broadly both systems comprise some conceptualisation, both have what we might refer to as an "operating" systems, both have some form of "information repository", and both have an application domain. In some cases the boundaries between these component are not as clear

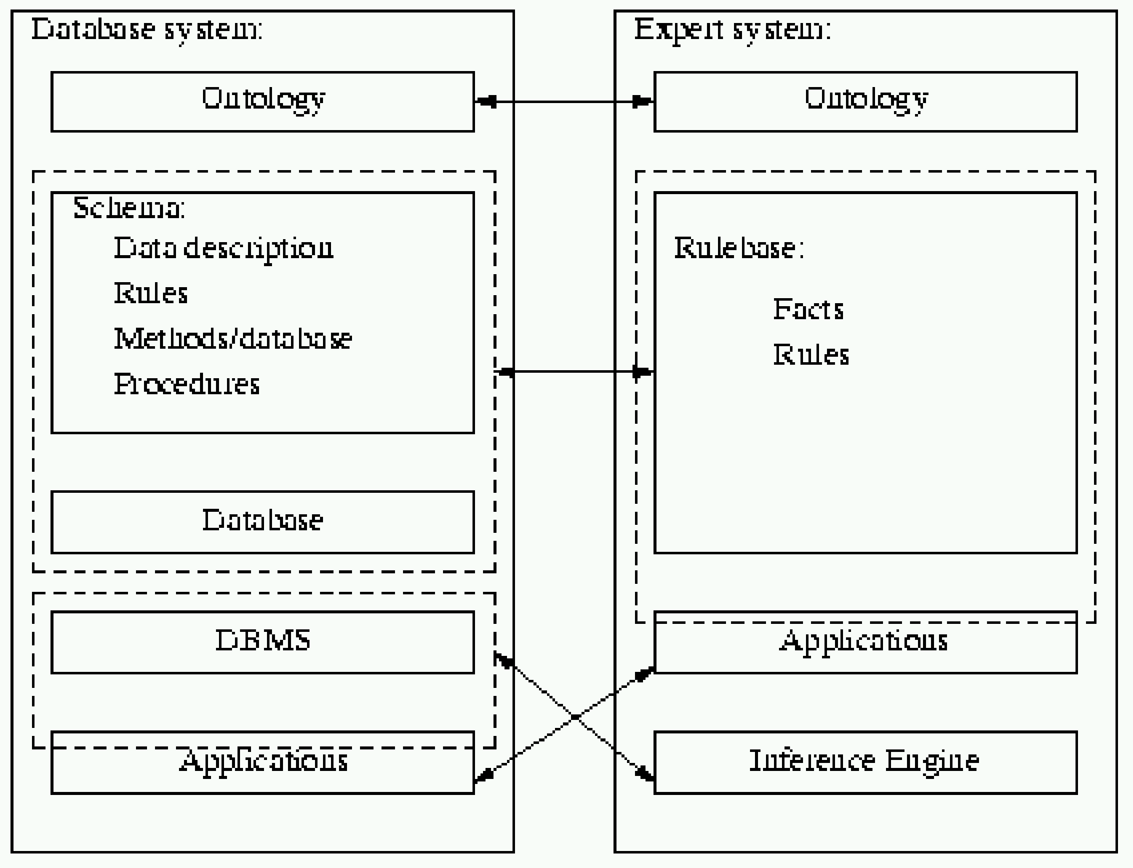

Figure 1: The mapping between DB and ES system components

as in other. This is particularly the case with respect to the application level. In the database community there is a definite approach of separating the database out from its applications (which may be varied), this distinction is not so clearly made in the context of expert systems that are often more closely linked to a key application. What ever the case, for discussion purposes, we can consider a hierarchy of:

- VVI **of** expert and database systems — the "operating" system level.
- VVI **in** expert and database systems, i.e. VVI **of** the knowledge/rule/data base — the "information repository" level and the main focus of this paper.
- VVI of expert and database systems applications

From the figure, at the conceptualisation level, in both expert and database systems, we have the concept of an ontology, i.e. a common understanding and vocabulary. In the case of expert systems an ontology is a set of definitions of; (hierarchically ordered) classes, objects, attributes, relations and constraints. It's main function is to provide a vocabulary for the expression of domain knowledge, i.e. a knowledge level description ([30]). A similar conceptualisation exists with respect to database systems — ontologies have been introduced for DBSs as a conceptual layer above the schema ([29]).

The "operating system" component — the Data Base Management System (DBMS) in the case of a database, and the inference engine in the case of an expert system — are traditional software systems in the sense that they are developed using "mainstream" software engineering tools and techniques.

The "information repository" in both cases provides the most fertile area of interest with respect to this paper. In the case of a database we distinguish between the database schema, which specifies the structure of the database; and the database instance, which specifies its actual content (Abiteboul et al. comment that the difference is analogous to the standard distinction between the type of a data item and its values as found in imperative programming languages). In the case of certain types of database further information in the form of rules and constraints may be added consequently there is some shared terminology. Inspection of the Figure also indicates that the concept of rules can appear in both systems, there is also a correspondence between "database facts" and "fact base data".

3.2 The relational model

From the above we can see that, in terms of architecture, there is a clear mapping between the two systems which if nothing else at least hints at the possibility of "cross-over" technology. A clearer overlap can be found if we compare the relational model (still the most prevalent DB model in industry) with rule base representations. The relational data model [10] represents data as tables of atomic values and defines an algebra for set-oriented associative manipulations. As note above it specifies two general integrity rules: *entity integrity* prohibits null-valued keys, ensuring that all data objects include an identifying name; and *referential integrity* prohibits foreign keys values which do not also occur as primary keys, thus avoiding *dangling references*. Relational databases ([17]) are the state of the art and benefit from the widely accepted APIs, ODBC and JDBC, and an international standard language, SQL, which provide the functionality of the relational algebra.

Given the above, identical relations can be specified as facts in a rule base which can the be queried in a similar manner to that provided by SQL, i.e. we can express a relational database using a clausal form identical to that used in rule bases. For example a PROLOG program can be viewed as a relational database where the specification of relations is partly explicit (facts) and partly implicit (rules). In addition most expert system environments provided mechanisms for updating, adding and removing facts (PROLOG provides the built in predicates `assert` and `retract` to do this).

3.3 Hybrid systems

A number of *hybrid* technologies have been developed which borrow from both the expert system and database domains. These include:

- Deductive databases
- Constraint databases
- Active databases,

all of which are supported by significant rule sets of a similar nature to that found in expert system rule bases. An overview of each of these hybrid models is presented below.

3.3.1 Deductive databases

If the relational model serves to bring the two technologies "in touch" the deductive model can be viewed as serving to merge the two technologies. Deductive database systems provide mechanisms for managing knowledge as well as data. They are a natural extension to the relational model. As such deductive databases are able to derive new facts using existing information explicitly stored in the database [32]. This is achieved by generalising the type of information that may be stored; in addition to simple facts it is also possible to store expert system style rule sets. This then produces a database system with similar properties to that of a logic programming environment. In some cases it is difficult to distinguish between what is a deductive database systems and a rule base system. The distinction between the two is that the primary concern of the first is the management of large amounts of data with the manipulation of that data as secondary [25]. Expert systems, on the other hand, are more concerned with the manipulation of "smaller" amounts of data and more complex amounts of knowledge, with the emphasis on manipulating that data in a manner that reflects the mode of working of a domain expert. We can also point to the mode of implementation and its manner of operation — DBMS or expert system language/environment, SQL or inference engine. The ratio of facts to rules may also be a useful indicator to distinguishing between an expert system and an expert database - an expert systems will (generally) contain many more rules than facts while an expert database system will contain many more facts than rules.

3.3.2 Active databases

Active databases support the automatic triggering of updates in response to internal or external events. This is generally achieved in an expert system like manner where forward chaining of rules is used to accomplish the update. Rules typically follow the ECA (Event-Condition-Action) format of:

```
on <event> if <condition> then <action>
```

The similarity between this and expert system style rules, especially in the case of production rule systems, is self evident. The distinction is that in a rule base the triggering event is user supplied rather than expressly included. In a sophisticated active rule base the action part of the rule may entail calls to further rules in a manner identical to that supported by expert system style rule bases. The overlap here is so close that the rule set that forms a component of an active database is generally referred to as a rule base [1].

3.3.3 Constraint databases

The constraint model ([24]) is an extension of research into Constraint Logic Programming (CAP) ([23, 21]) which itself is an extension of logic programming. In CAP constraint satisfaction is embedded into a logic programming environment/language, for example Sicstus PROLOG which is an extended version of traditional PROLOG which incorporates CLP features.

The basic idea of constraint databases is to replace the notion of a tuple in a relational database by that of a generalised tuple, i.e. a conjunction of constraints. For example given a tuple:

$$(a_1,\ ...,\ a_n)$$

this can be regarded as a generalised tuple of the form:

$$(x_1 = a_1) \wedge ... \wedge (x_n = a_n)$$

Of course this model also needs to be supported by an appropriate "constraint database" query language. The extension of both expert system and data base models to incorporate constraint satisfaction features may be viewed as another growing area of overlap.

It should also be noted here that the use of constraints is a recognised as an important tool for database integrity checking. In early work in this area constraints were expressed in terms of clausal logic. More recently much more sophisticated constructs have been adopted.

3.4 Emerging database models

The emerging third generation of DB technology is predominantly object and object-relational. Respective de facto standards, i.e. ODMG [8] and the SQL3 standard proposals [4], can be viewed as object-oriented programming languages with added database facilities and relational databases systems with added object-oriented features. Both are technology (rather than theory) driven and have emerged in response to the realization that the application of relational systems is impeded by a limited type system. Speculative requirements for these technologies were set out at the turn of the decade in the object-oriented and third generation database manifestos [5, 36].

The current predominance of object-oriented programming (Java, Etc.) has also led to the extension of expert systems into the OO domain ([34]). Much of this work is speculative, and thus it is difficult to say where the connection between OO databases and OO expert systems will be, however, it should not be forgotten that frame based expert systems were a feature of the 1980s (many written in LISP) and that the frame concept is not that different from the ideas associated with classes and instance of classes.

3.5 Summary

From the above we can identify a number of areas of overlap. We can summarize these as follows:

- There is a direct mapping between the architectural components of both types of system.
- Both systems have an underlying conceptualisation associated with them, which can be expressed using some common vocabulary using (i.e. the ontology concept).
- There is a clear overlap between the relational model and the clausal forms used in many rule based systems.
- Hybrid systems contain components borrowed from both domains.
- There is a close relationship between the validation and verification of the DBMS and Inference engine, but this is outside the scope/spirit of VVI as envisaged here.

We can also say that as soon as a rule base system starts to become dynamic, i.e. evolving with time, integrity becomes an issue.

A word of warning is also appropriate here, the relational model remains the industry standard, the remaining models at best are still regarded as experimental and at worst as "academic toys". Whether any of the alternative models will establish a firm place in industry remains a topic for debate and "wait and see".

4 FUTURE DIRECTIONS (AND "CROSS-OVERS")

In this section we consider a series of possible directions for future work. Some of these are directed at the areas of overlap identified above, others focus on some additional areas which may come under the remit of "cross-over" technology.

4.1 Conceptualisation

The ontology component is "an explicit conceptualisation of the domain" ([7]). Bench-Capon ([7]) has identified a number of areas where ontologies which conceptualise the problem domain can be used to support V&V of expert systems ([7]). It is suggested that the added coherence of an explicit domain conceptualisation gives a more motivated and focussed role for experts who are "signing off" a KB, reduces the scope for subjective interpretation which can mask errors, and provides the possibility of a test harness. More specifically, an ontology provides an agreed representation of the universe of discourse and hence support for validation and verification based upon real world semantics. Although Bench-Capon admits that an ontology is at present not usually defined for expert systems it can be argued that there are advantages in doing so. A similar case can be put for systems for database systems. It is likely that some advantage can be gained from the application of the ontologies concept for V&V at the conceptualisation level. There are already a number of practitioners carrying out V&V studies for expert systems at the conceptualisation level ([19]) although the term "ontology" is not explicitly used.

4.2 Relational models

The correspondence between schema/database and rule base is most apparent for relational databases, since they share a common theoretical foundation with expert systems, i.e. clausal form. This has been exploited, for example in deductive databases where relational DBMSs provide back-end fact storage. However, this correspondence has not been exploited practically to any extent for V&V of database systems or for integrity maintenance of evolving expert systems.

Given that a relational database can be represented in rule base terms (subsection 3.2) it should be possible to implement classic database integrity checking techniques in these terms. It may be that the "intelligent" features of these systems can then be used to enhance the "productivity" and capabilities of these techniques.

4.3 Hybrid systems

Although a correspondence can be established between relational database systems and expert systems the correspondence is much stronger in the context of the hybrid models that contain significant expert system components. For example the rule sets associated with deductive and active database systems are subject to identical anomalies and symptoms of errors that the expert system's V&V community have been concerned with for many years. Some work is already being undertaken in this area with respect to active databases. For example Aiken et al. ([3]) have suggested methods to identify infinite loops in active database rule sets using an approach known as the *trigger graph method*; similar work has also been undertaken by Lee and Ling ([26]).

4.4 The object-oriented model and frame based systems

The predominant expert system methodology is currently founded on the rule base approach, the alternative frame based approach has become unfashionable, instead this has been replaced by work on object oriented expert systems. For example Riley's CLIPS systems ([34]). To the best knowledge of the authors of this paper no work has been done with respect to V&V of object oriented expert systems; it is therefore not possible to make any predictions concerning the possibilities of "cross-over" in this context although it might be an interesting area for future investigation.

4.5 The application of database techniques to the expert system V&V domain

In the foregoing we have concentrated on areas where current V&V techniques as applied to expert systems may find a role on the context of various database models, especially where those models intersect with the expert system domain. However, at least in theory, the opportunity for crossover should be "two-way". For example if a relational database can be represented in a clausal form, it should be possible to represent a rule and fact base as a set of database tables. This is undoubtedly true but does it "buy" you anything? With respect to inferencing strategies probably not, but in the context of V&V this may well offer advantages. Support for the latter conjecture can be found in the many tabular techniques used by expert system V&V practitioners; truth tables ([38]), incidence matrices ([2]) and other tabular representations ([27]).

Another area where one might expect the application of DB theories to ESs (and by extension DDBs) is design theory. Normalisation is a formal approach to the refinement of relational DB design. (by ensuring no unnecessary replication of data). Normalisation, together with representational theory [28], provides a theory of "goodness of relational DB design". Again one would expect this to improve our ability to design good representations of facts within ESs (and DDBs). However, that is not the case, since normalisation is necessary for relational databases only because of the 1NF constraint which prohibits direct representations of complex objects. In DDB and ESs, complex structures can be modeled. For example, a list can be accommodated in a DDB fact (although physical storage may be harder if a relational DB is actually used for the fact storage).

4.6 Other possible opportunities

There are a number of areas of research that are currently under consideration within the expert systems V&V community which, although not directly applicable to the overlap identified here, have possibilities with respect to their application to database systems. Of note is Vermesan's work on the classification of KBS ([39]). Knowledge Base (KB) certification uses a number of different V&V methods such as static analysis, testing, inspection, and modeling to assess the quality of the KB component. This work is currently "on going". Although there is a perceived need for the certification of KBS (by for example marine classification societies such as Det Norske Veritas) does this also apply to database systems? and, if so how should this be undertaken?

A further consideration is that expertise is required by DB administrators and users, and thus expert system techniques may be appropriate to improve the quality of usage of DB technology by the non-skilled operator. Similar considerations may be appropriate to "intelligent querying".

Finally, although we have noted that the VVI of DBMS and inference engines is a subject for traditional software engineering techniques, a test suite of rules and or facts to test this level might be an attractive proposition, possibly providing a validity benchmark to put along side the performance benchmarks used with DBMSs. Clausal form providing a lingua franca across systems. The varied ways queries may be executed by a DBMS make this a more complex operation than might be at first assumed for database systems or hybrid systems that use DBMSs for fact or rule storage. Since most DBMSs use a query optimiser the way a query is executed may depend on a large number of factors, such as size of relations and statistics on that data, and its correct evaluation in all circumstances would be hard to test or guarantee. Errors in a B-tree algorithm might only come to light for example if an existing relation had its structure changed. [1]

5 CONCLUSIONS

From the foregoing it is clear that there is currently little common research effort or shared results between the database and expert systems communities, despite the view held by members of both communities that dialogue and collaborations is long overdue. It is hoped that some of the ideas outlined in this paper will encourage this collaboration and that this in turn may result in some genuine technological advances that will benefit all concerned.

Some work has already been undertaken under the heading of VVI. The 1st International Workshop on Validation, Verification and Integrity Issues (VVI'98) to place in Vienna last year ([40]); a review of this workshop can be found in Bench-Capon et al. ([6]). It is intended to hold a second VVI workshop, in conjunction with DEXA, in the year 2000.

References

[1] Abiteboul, S., Hull, R. and Vianu, V. (1995). Foundations of Databases. Addison Wesley, Reading, MA.

[2] Agrawal, R.and Tanniru, M. (1991). A Petri-net Based Approach for Verifying the Integrity of Production Systems. Knowledge-Based Systems Verification, Validation a nd Testing Workshop Notes, AAAI'91, Anaheim CA.

[3] Aiken, A., Widom, J. and Hellerstein (1992). Behavior of Database Production Rules: Termination, Confluence and Observable Determinism. Proceedings ACM SIGMOD, International conference on the Management of Data, pp59-68.

[1] As an aside DBMSs have the notion of serializability in transactions (i.e. ''isolation" — the I in ACID test) that aims to the same consistent end point no matter what the route through. Is there an expert system counterpart for rule firing?

[4] ANSI, (1998). SQL3 drafts and discussion documents can be found at ftp://speckle.mcsl.nist.gov/isowg3/dbl.

[5] Atkinson, M.P., et al., (1990). The Object-Oriented Database System Manifesto, in Kim, W., and Nicolas, J-M., and Nishio, S. eds., Deductive and Object-Oriented Databases, pp 223-239.

[6] Bench-Capon, T., Castelli, D., Coenen, F., Devendeville-Brisoux, L., Eaglestone, B., Fiddian, N., Gray, A., Ligeza, A. and Vermesan, A. (1999). Validation, Verification and Integrity Issues in Expert and Database Systems. Expert Update, Vol 2, No 1, pp31-35.

[7] Bench-Capon, T. (1998). The role of Ontologies in the Verification and Validation of Knowledge based Systems. In Wagner R. R, (ed), Proceedings DEXA 98 Ninth International Workshop on Database and Expert Systems Applications, IEEE Computer Society, pp64-69.

[8] Cattell, R. and Barry, D.K. (eds) (1997). The Object Database Standard: ODMG 2.0. Morgan Kaufman, San Fransisco.

[9] Coenen, F. and Bench-Capon, T. (1993). Maintenance of Knowledge Based Systems: Theory, Tools and Techniques. Academic Press, London.

[10] Codd, E.F. (1970). A Relational Model for Large Shared Data Banks, Communications of the ACM,13(6), pp 377-387.

[11] Codd, E.F. (1979). Extending the Relational Database Model to Capture More Meaning. ACM TODS 4 4 Dec.

[12] Codd, E.F. (1985). Is Your DBMS really relational? Computerworld, Oct 14.

[13] Cohen, J. (1990). Constraint Logic Programming Languages. Communications of the ACM, Vol 33, pp52-68

[14] Cragen, B.J. and Syeudekl, H.J. (1987). A Decision-Table-Based Processor for Checking Completeness and Consistency in Rule-Based Expert Systems. International Journal of Man-Machine Studies, Vol 26, pp633-648.

[15] Craw, S. (1998). KRUSTWorks: Developing a Generic Refinement Toolkit. Expert Update, Vol 1, No 2, pp35-47.

[16] Date, C.J. (1986). An Introduction to Database Systems. Vol 1 4th ed, Addison-Wesley.

[17] Eaglestone, B. (1991). Relational Databases. Stanley Thornes.

[18] Eaglestone, B. and Ridley, M. (1998). Object Databases: A Introduction. McGraw Hill.

[19] Fensel, D. and Schonege, A. (1997). Specifying and Verifying Knowledge-Based Systems with KIV. Proceedings EUROVAV'97, Katholieke Universiteit Leuven, pp107-116.

[20] Ginsberg, A. (1988). Knowledge Base Reduction: A New Approach to Checking Knowledge Bases for Inconsistency and Redundancy. In Proceedings AAAI'88, St. Paul, MN, Vol 2, pp585-589.

[21] Jaffar, J. and Maher, M. (1994). Constraint Logic Programming: a survey. Journal of Logic Programming, Vol19-20, 503-581.

[22] van Harmelen, F. and ten Teije A. (1997). Validation and Verification of Conceptual Models of Diagnosis. Proceedings EUROVAV'97, Katholieke Universiteit Leuven, pp117-128.

[23] Van Hentenryck, P. (1989). Constraint satisfaction in Logic Programming. Cambridge, MA: MIT Press.

[24] Kanellakis, P.C., Kuper, G.M. and Revesz, P.Z. (1995). Constraint query languages. Journal of Computer System Science, Vol 51, No 1, pp26-52.

[25] Kerschberg, L. (Ed.) (1986). Expert Database Systems: Proceedings from the First International Conference. Benjamin-Cummings.

[26] Lee, S.Y and Ling, T.W. (1997). Refined Termination Decision in Active Databases. In Hameurlain, A. and Tjoa, A.M. (Eds), Database and Expert Systems Applications, (Proceedings DEXA'97), Lecture Notes in Computer Science 1308, Springer Verlag, pp182-191.

[27] Ligeza, A. (1998). Towards Logical Analysis of Tabular Rule-Based Systems. In Wagner R. R, (ed), Proceedings DEXA 98 Ninth International Workshop on Database and Expert Systems Applications, IEEE Computer Society, pp30-35.

[28] Maier, D. (1983). The Theory of Relational Databases. Pitman Science Press, Maryland.

[29] Masood, N. and Eaglestone, B. (1998). Semantics Based Schema Integration. In Quirchmayer, G., Schweighofer, E. and Bench-Capon, T.J.M. (Eds), Database and Expert Systems Applications, proceedings DEXA'98 (Vienna), Lecture Notes in Computer Science 1460, Springer Verlag, pp80-89.

[30] Newell, A. (1982). The knowledge level. Artificial Intelligence, 18, pp87-127.

[31] Nicolas, J-M. (1982). Logic for Improving Integrity Checking in Relational Data Bases. Acta Informatica 18, pp 227-253.

[32] Paton, N., Cooper, R., Williams, H. and Trinder, P. (1996). Database Programming Languages. Prentice-Hall.

[33] Preece, A.D. and Shinghal, R. (1991). COVER: A Practical Tool for Verifying Rule-Based Systems. Knowledge-Based Systems Verification, Validation and Testing, Workshop Notes AAAI'91, Anaheim, CA.

[34] Riley, G. (1999). CLIPS — A tool for building expert systems. `http://www.ghg.net/clips/CLIPS.html`

[35] Rousset, M-C. (1997). Verifying the World Wide Web: A Position Statement. Proceedings EUROVAV'97, Katholieke Universiteit Leuven, pp95-104.

[36] Stonebraker, M., et al. (1990). The Committee for Advanced DBMS Function. Third Generation Database System Manifesto, SIGMOD Record, 19(3), 1990.

[37] Suwa, M., Scott, A.C. and Shortliffe, E.H. (1984). Completeness and Consistency in a Rule-Based System, In Buchanan, B.G. and Shortliffe, E.H. (eds), Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project, Addison-Wesley.

[38] Vanthienen, J. and Dries, E. (1993). Illustration of a Decision Table Tool for Specifying and Implementing Knowledge Based Systems, Proceedings 5th Int. Conf. on Tools with AI, pp198-205.

[39] Vermesan, A.I. (1998). Software Certification for Industry — Verification and Validation Issues in Expert Systems. In Wagner R. R, (ed), Proceedings DEXA 98 Ninth International Workshop on Database and Expert Systems Applications, IEEE Computer Society, pp3-14.

[40] Wagner R. R. (Ed) (1998). Proceedings DEXA 98 Ninth International Workshop on Database and Expert Systems Applications, August 260-28, 1988, Vienna, Austria. IEEE Computer Society, 1998.