From Data Independence to Knowledge Independence: An on-going Story

Laurent Vieille
Next Century Media Inc., 12 Avenue des Prés,
78 180 Montigny le Bretonneux, France.
e-mail: lvieille@computer.org

Abstract

It is today widely accepted that "Business Rules Independence" is required for information systems to better and more rapidly adjust to changes in the business environment. This paper[1] attempts to articulate how logic based database systems provide adequate technology for better "Business Rules Independence". These systems do so by going beyond "Data Independence" and by providing "Knowledge Independence". This paper benefits from the experience gained in developing and marketing the VALIDITY deductive and object-oriented database system during the last few years. Current applications and specific data management techniques to be used are discussed.

1.0 Introduction

While research on logic-based database systems was fashionable in the 1980s, recent years have witnessed a decrease of interest towards this formalization of databases. Several reasons have been put forward to justify such a decreasing interest. First, the proximity of the relational model with logic database formalizations makes it possible, at least in theory, to extend SQL engines with features uniquely conceptualized in logic. Given the commercial importance of SQL products, it appeared to a number of researchers and of practitioners as being the approach to pursue. It is indeed a common opinion that, while researchers benefit from using languages such as Datalog to study semantic and algorithms, their results should be translated into SQL to be exploited in the commercial world. Second, from the application point of view, the interest of logic-based database systems had remained unclear. Third, robust commercial implementations of such systems were missing.

At the same time, new data management requirements are appearing on the market place. Some requirements are related to the Web: federation of data sources, management of semi-structured documents, etc. Other requirements are coming from a generic need known by practictioners as: "*Business Rules Independence*". This concept reflects the fact that information systems must become more and more reactive to changes in business environments. All the components of information systems (client, middle-tier, server) must contribute to the support of such a generic concept, and this paper addresses the role of data management systems in this new dimension of information systems.

More specifically, this paper attempts to articulate how logic-based rule & data base systems contribute to "Business Rules Independence". Indeed, while it is clear that no single technology will provide the complete solution for such a wide-ranging issue, there are reasons to believe that such systems are particularly appropriate to "deal with changes". (See also [9.], [6.], [14.].)

These aspects are reviewed below in the light of the experience gained within the VALIDITY project during the last five years. VALIDITY is a DOOD (Deductive and Object-Oriented Database) product capitalizing on earlier research efforts, in particular on the deductive technology developed at ECRC in the late eighties (See [1.][2.][12.]). Initially developed at BULL, it is now further developed within Next Century Media (NCM). This product is currently shipping as an embedded system within Opti*Mark, an NCM application for addressable advertising [22.].

Besides this introduction, this paper is organized in 4 sections, each section containing both a presentation and a critical review. Section 2 presents a broad view of "Business Rules Independence" and of "Knowledge Independence" as achievable by logic-based data base systems. Section 3 and 4 respectively discuss applications and technology aspects. Section 5 is the conclusion.

Proceedings of the 24th VLDB Conference
New York, USA, 1998

2.0 From Data Independence to Knowledge Independence

2.1 Presentation

In today's quickly evolving world, the reactiveness of a company is becoming one of its competitive advantages. The ability to quickly adjust an offering and an organization can win new markets to a company.

On the organization side, this requirement for more flexibility is part of the motivation for business process reengineering. On the technical side, flexibility requires quick adaptations of information systems; it is widely perceived that "*Business Rules Independence*", i.e., the independence between the rules governing a company's operations and its information system, is the right path to improve the ability to deal with changes.

As this notion can relate to many different aspects of information systems, we use a specific expression, "*Knowledge Independence*", when talking of server systems integrating data and logic-based rule management in a seamless manner. Thus, Knowledge Independence can be seen as a technical version of Business Rules Independence.

From the database point of view, Knowledge Independence corresponds to a quantum step forward beyond "Data Independence"[1]. Indeed, Knowledge Independence extends Data Independence with the ability to manage rules governing the data, independently from the applications. (See [9.] for an introduction to Knowledge Independence, [11.],[13.] for related works).

FIGURE 1.. Data vs. Knowledge Independence

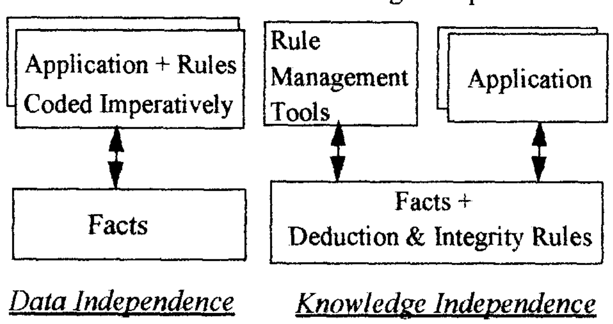

Figure 1 above illustrates how Knowledge Independence brings advantages in building and maintaining information systems:

1. *factorization*, i.e., the ability to share deduction and integrity rules between multiple applications, facilitates application development;

2. *modularity*, i.e., the ability to add, delete or modify rules, even if the predicate they define is referenced by other rules and by applications, facilitates application maintenance;

3. *rule management tools* provide a more abstract and more powerful way to manage an essential part of the information system.

In this context, logic-based systems provide clear benefits. Declarativity allows the automatic optimization and easy modification of the rule base, without having to modify the applications. Predicate logic provides an unmatched wealth of algorithms and inference techniques for the development of rule management tools with unique features: sample database browsing, consistency checks, explanations, etc.

Those benefits are limited, as in any other software technology, by theoretical results. Extended Datalog is a query language enabling query optimization but it does not have the full power of a programming language. Similarly, automatic checking tools face a number of semi-decidability or undecidability theoretical results. To face those difficulties, pragmatic approaches must be chosen, some of which are outlined in Section 4.0.

2.2 Discussion

A first question is how the logic-based approach relates to other technologies which also improve the flexibility of information systems. In object-oriented languages, methods can be modified or overwritten without changing the signature of a class; in component-based development, components can be replaced without touching applications using them. Actually, the benefits of these three technologies are complementary: The key benefits of rule-based approaches (declarativity, rule-management tools) are not provided by main-stream object-oriented tools or component-based programming.

A second question, is how providing Knowledge Independence in the server compares to building rule-based components in other tiers of the information system: in the client, e.g., using expert system shelves or at the middle-tier, by means of adequate middleware. Rule-management at the client is usually limited to domain checks and to user interface event programming; items which are complementary to those described under "Knowledge Independence". However, the approach integrating rule management with data management may have to compete with middle-tier components (based on Prolog or on other rule languages) coupled with a DBMS. This is discussed in Section 4.2 on page 4.

A third question is whether the various elements of Knowledge Independence can be achieved by extending SQL systems. Indeed, the SQL language and systems supporting it can theoretically be extended to match the expressive power of Datalog or to provide more adequate modularity than in the current SQL standard[2] and to enable factorization.

However, SQL has not been designed as a knowledge representation language, and it is poorly suited for the direct support of knowledge management tools. Indeed,

1. I.e., the ability offered by a system both to define the data structure and to manage data independently from the application itself.

2. which forces the deletion of a view signature and of items depending on it, before this definition can be changed.

the ability to support rule management tools underlines the fact that Knowledge Independence is more than a mere factorization of rules out of the applications. Without such tools, the attractiveness of Knowledge Independence would be much smaller. It is likely that expressions factorized in the data base would remain at the ad-hoc query level, as seen in most relational applications today.

3.0 Applications

3.1 Generic Applicability

A rather general investigation of applications that would benefit from logic-based database systems was conducted in the early years of the VALIDITY project. Knowledge Independence appeared to be beneficial to a number of novel applications domains, as summarized in [9.].

For instance, highly-regulated areas such as the management of hazardous goods, would benefit from a declarative expression of their regulations. Concurrent engineering applications would take advantage of integrity rules setting up the rules of cooperation.

The definition of a few criteria to qualify candidate applications also emerged from this initial phase. These criteria are:

1. management of reasonably-sized data and rules;
2. rules relate to data management or semantic;
3. frequent changes in the rule set.

Although it is natural that a new system focuses on novel applications, this exploration work has also shown that more traditional business applications would also benefit from rule management.

This analysis is supported by reports published in the recent years by market and technology analysts. Patricia Seybold in [3.] insists on the urgent need for most business applications to be able to manage their rules dynamically, outside application programs, but while keeping an efficient interaction with their associated data. IDC in [4.], while analyzing the needs for electronic commerce services, insists on the importance of content servers whose main characteristics are precisely those of DOOD systems. Finally, in various reports such as [5.][6.][7.], the Garner Group underlines the requirement for an efficient integration between rule management techniques and data managers in order to correctly fulfill market needs.

3.2 Two Particular Examples

NCM is currently shipping VALIDITY as an embedded system within the Opti*Mark application [22.]. Opti*Mark is an application for handling addressable advertising, in the area of new television infrastructure. Indeed, in television broadcast systems as we know them today, advertisers have few means to target their ads to a specific audience. While better targeting has been in place via other media for years (direct mailings, specific editions of magazines, etc.), this is going to be possible on television systems only using emerging infrastructures, such as digital set-top box, two way connections, etc.

In this context, Opti*Mark handles advertisers targeting criteria, viewer information, program schedule and information, and generates recommendations on which ads should be shown to whom. Opti*Mark relies on VALIDITY to exploit advertising strategies, expressed as rules, in order to rank the ads according to viewers profiles. Using declarative rules to express both in-house advertising expertise and specific advertisers' strategies is a definite competitive advantage, as it allows to quickly adjust to market or strategy changes.

The following rule, extracted from the actual Opti*Mark application, expresses the fact that ads from the campaign "Cp" shall not be shown in episodes of the series "S" if there exists a subject "Sub" of series "S" which was explicitly excluded by the advertiser:

```
adExSeries(camp:Cp, series:S) <-
  adSubject(camp:Cp, subj:Sub, wght:0) and
  seriesSubj(series:S, subj:Sub);
```

The following rule, used for demonstration purpose, shows the reactivity of the system. In this case, because of a hurricane announcement, priority of showing ads about "Florida" to people not living in "Florida", is divided by 2:

```
newrating(pers:P, ad:A, or:Old, nr:New)<-
  isAbout(ad:A, subj:'Florida') and
  not viewer(pers:P,state:'Florida') and
  New == Old / 2
```

Another current application is devoted to the computation of financial ratios for investment analysts. In this application, the ability to define new ratios by means of additional rules, without changing the interface, enables the customization of the application to each analyst.

3.3 Discussion

In the two applications mentioned above, the essential benefit brought by VALIDITY is the *modularity of rule declaration*; this allows, in the first case, an impressive reactivity to environment changes and, in the second case, an easy customization of the ratios to each analyst.

As another interesting benefit, let us note that *clarity of the language semantics*, in particular of its aggregate operators in comparison to Prolog, has played a significant role in convincing some technical people within a customer organization. This element is of course subject to subjective assessment and its economical value is usually less apparent to management than the first type of benefit.

While the ability of VALIDITY to check integrity constraints is perceived as an appealing feature by users, this ability is not demonstrated in the two applications above where the data is cleaned a priori. It is anticipated that this feature will be used in interactive applications as those described in the previous section.

Finally, performance is a key aspect when it comes to handling large data sets. While VALIDITY does not compare yet with relational products which have been optimized during 10+years, users trust that the integrated architecture is scalable and will allow gradually improved performance.

4.0 Technology

4.1 Presentation

In this section, we discuss the technology necessary for systems supporting Knowledge Independence, together with the robustness and scalability of current data management systems. We try to isolate those technologies which go beyond traditional data management technologies, and more specifically beyond data models, query evaluation and integrity checking algorithms studied in the 1980s (see [1.] for an overview) which are widely available.

The data model and manipulation language of a DOOD system must be able to support declarativity, modularity, factorization for both deduction and integrity rules and advanced rule management tools; it must include generally accepted notions of data models in order not to go backwards: typing, inheritance, etc.

Enough basic research results were found to be available when designing the DEL language ("Datalog Extended Language") for VALIDITY. DEL incorporates an ODMG-like object model [16.], with fact identity and complex values, with Datalog; extended with negation, grouping and quantifiers, to express deduction and integrity rules. In addition, specific care was put in the support of 'unknown values', a logical version of SQL NULLs. Similarly, to cope with requirements going beyond extended Datalog, DEL includes a procedural language and incorporates a foreign function capability to implement DEL functions and predicates in languages such as C and C++. This provides a functionality similar to data blades. Presentations of DEL can be found in [9.], in [15.] and in the specific chapter devoted [11.].

Basic algorithms to evaluate (recursive) deductive queries and to check integrity have been made available by research in the 80s. From this point of view, the VALIDITY deductive engine technology derives from the ECRC EKS system [12.]. However, more techniques are needed to improve the scalability of cost-based optimization in the presence of large rule sets.

The storage manager of a DOOD system has to provide the traditional persistency, transactional and recovery functions. In addition, it must fulfill several specific requirements: it must provide both fast scan (typical of relational systems) and fast navigation (typical of object-oriented systems); it must support both fact (object) identity and complex values. Finally, its concurrency control protocol must face a specific issue, related to the need of a sound verification of general integrity rules. Two-phase locking suffers from the phantom issue when used at the page or record granularity, which makes it unable to guarantee full transaction isolation. Predicate or table locking is not practical in the general case.

VALIDITY incorporates a concurrency control protocol specifically merging locking and version management. This protocol, similar to the one reported in [24.], developed independently, is under validation jointly with INRIA [23.].

Rule management tools are essential for the actual support of "Knowledge Independence". Besides state-of-the-art graphical interface to enter, browse and modify rules independently from applications, rule management aspects appear in various tool areas:

1. *Debugging.* A facility which naturally comes to the mind of AI experts, is the possibility to provide explanations on deductions or on deduction failures. This aspect becomes particularly important when the data set grows and the rule set becomes complex: The difficulty of providing meaningful explanations in such an environment makes the realization of an explanation tool a non-trivial task, and this is the subject of a current study (see [20.]).
2. *Methodology.* Only recently has appeared the first effort to incorporate rule aspects in design methodologies [11.]. More work is needed.
3. *Design Tools.* Inference techniques available for declarative logic expressions make it possible to study rule sets at the design stage. The best known such facility consists in checking whether the rule set is consistent. However, it also appears that designers need to check the semantics of their rule set by browsing through sample models of their rules. Efforts in this area are represented by the VALIDITY schema checker prototype [17.], extending SATCHMO with equality and arithmetic predicates [18.][19.], and by on-going research work; e.g., at Munich University [26.]. Such tools clearly represent an improvement over current design tools, where much of the semantic remains informally specified.

4.2 Discussion

Coupling of inference engines such as *Prolog* with RDBMs is often seen as an alternative to the integration of data and rules. This low-cost approach has many shortcomings. Prolog is not purely declarative and lacks several aspects of data manipulation languages (aggregates, clean updates, integrity rules, etc.). The lack of integration between rules and data is an obstacle to scalability and robustness: rule optimization is difficult without an access to physical data structures, concurrency control and recovery issues suffer from the coupling. Thus, Prolog applications or other rule-base applications should be seen as middle-tier components contributing, in their specific way, to "Business Rules Independence".

Another approach to rules in databases is the research around *active (or production) rules*. In this approach, an Event-Condition-Action (ECA) model is adopted to extend the so-called "trigger" facility of RDBMs. The need of "active databases", i.e., of databases able to react when certain conditions are met, has been identified in several applications studied in the context of VALIDITY.

However, the equation "*active databases=production rules*" should be evaluated with care: (1) production rules systems have proved to be difficult to maintain, for instance in the case of expert systems; (2) their lack of declarativity is an obstacle to automatic optimization and to powerful rule management tools. As a consequence, their contribution to true "Knowledge Independence" has to be assessed with care.

Thus, we feel that other approaches to "active databases" have to be investigated. For instance, [25.] proposes an automatic enforcement of integrity rules, while maintaining the true declarativity of their expression. Similar approaches must be studied to allow databases to perform an external action when a condition is met.

Finally, let us list some of the feedback obtained from application development: foreign functions were designed as an answer to an application requirement; the need for a debugging tool including explanations is dearly felt; more materialization facilities are needed; complex values have proved important for the migration of audience data from a COBOL application to VALIDITY.

5.0 Conclusion

In this paper, we have argued that pursuing the path from "Data Independence" to "Knowledge Independence" in logic-based data base systems will provide advances in the trend towards more flexible, more functional and easier-to-maintain information systems.

In the light of the experience gained with VALIDITY, we have indicated that several of the gains expected from the initial, broad picture were actually met in practical applications, while others remain to be validated via more tools and more applications.

We have also given indications on what areas of data management technology need to be developed to meet the promises outlined initially.

Acknowledgments. The author thanks B. Bergsten, G. Despain, L. Fournié, Ph. Jennequin, A. Lefebvre, B. Wappler for their contributions.

References.

1. *"Summary State of the Art on Deductive and Object-Oriented Databases"*, The VALIDITY Team. April 1996.

2. *"Deductive Object-Oriented Databases. Technology, Products and Applications: where are we?"*, Invited Paper; J.M. Nicolas, Int. Symp. on Digital Media Information Base (DMIB'97). Nara, Japan, Nov. 1997.

3. *"Notes on Information Technology: Do you know where your rules are?"* Patricia Seybold, Computer World, Oct. 4th, 1993.

4. *"The Electronic Commerce Services Market Review and Forecast"*, IDC report #9627, March 1995.

5. *"Essential Technologies for 1995"*, Gartner Group, T-ATT-311, Feb. 1995.

6. *"Rules in Databases: New Life for Tainted Technology"*, Gartner Group, T-517-229, Dec. 1993.

7. *"Business Process Automation"*, Gartner Group, R-800-116, February 1993.

8. *"Foundations of Databases"*, S.Abiteboul, R.Hull and V. Vianu, Addison-Wesley, Reading, MA, USA, 1995.

9. *"Applications of Deductive-Object Databases using DEL"*, O.Friesen, G.Gauthier-Villars, A.Lefebvre and L. Vieille. In [10.].

10. *"Applications of Logic Databases"*, R. Ramakrishnan (Editor), Kluwer Academic, Boston, MA, USA, 1995.

11. *"Designing Database Applications with Objects and Rules"*, S. Ceri and P. Fraternali, Addison Wesley, 1997.

12. *"Architecture and Design of the EKS Deductive Database System"*, L. Vieille, P. Bayer and A. Lefebvre, BULL-ECRC Technical Report, 1993.

13. *"Deductive Object-Oriented Programming for Knowledge-Base Independence"*, Y.Yanagisawa, M.Tsukamoto, S.Nishio. 4th Int. Conf. on Deductive and Object-Oriented Databases (DOOD) Singapore, Dec.1995.

14. *"Data Dredging"*, S.Tsur, Data Engineering, Dec. 90.

15. *"The DEL Language Reference Manual"*, V1.3, the VALIDITY Team, NCM, March 1998.

16. *"The Object Database Standard: ODMG 1993"*, R.G.G. Catell (Ed.), Morgan Kaufman, 1994.

17. *"VALIDITY Schema Checker"*, C.Fournet,NCM, 1994.

18. *"SATCHMO: A Theorem Prover Implemented in Prolog"*, R. Manthey and F. Bry, Proc. 9th Int. Conf. on Automated Deduction (CADE), May 1998.

19. *"E-SATCHMO: Introduction de l'égalité dans le démonstrateur automatique SATCHMO"*, L. Herr, Master Thesis, Ecole Normale Supérieure, Sept. 1993. In French.

20. *"A Set-Oriented Meta-Interpreter driven by a Relational Trace for Deductive Database Debugging"*, S.Mallet and M.Ducassé, "Logic-based Program Synthesis and Transformation", LOPSTR'98, Manchester, June 1998

21. *"A Deductive and Object-Oriented Database System: Why and How"*, L. Vieille, ACM-SIGMOD, June 1993.

22. *"VALIDITY: Knowledge Independence for Electronic Mediation"*, Invited Paper, L. Vieille, Practical Application of Prolog and Constraint Technology (PAP/PACT98), London, March 1998, G.A. Narboni, Ed.

23. Laurent Fournié, PhD thesis in preparation, 1998.

24. *"Techniques pour la Conception et l'Exécution efficace de Transactions."*, D. Tombroff, PhD thesis, Université Paris VI, June 1996.

25. *"Active VALIDITY"*. J.A. Fernandez; Internal VALDITY Report. 1996.

26. *"SIC: Satisfiability Checking for Integrity Constraints"*, F.Bry, N.Eisinger, H.Schütz, S.Torge, Report PMS-FB-1998-3, L. Maximilian Universität, Munich.