Expressing Business Rules using Object Role Modelling

Adrian Hargreaves
Department of Information Systems,
Massey University, Wellington
a.j.hargreaves@massey.ac.nz

Numerous industry surveys have suggested that most IT projects still end in failure. Incomplete, ambiguous and inaccurate specifications are cited as a major causal factor. Traditional techniques for specifying requirements most often lack the expressiveness with which to model subtle but common features within organisations. As a consequence, many of the business rules that determine the structure and behaviour of organisations are simply not captured until the latter stages of the development lifecycle.

Business rules originate as policies that are adopted by organisations to achieve their higher-level goals. The Business Rules Group (BRG) has defined a detailed taxonomy for these rules. The concepts and definitions within this taxonomy can guide analysts to seek appropriate information from end-users to define their requirements. Unfortunately, most techniques that are currently employed for representing requirements incorporate implementation details unfamiliar to business people. Unless domain experts can actively challenge the analyst's understanding of these rules, how can analysts be certain they are defining the required system?

A Fact-based technique called Object Role Modelling (ORM) has been investigated as an alternative approach, and a case study conducted by the author has suggested that it could provide a mechanism for improving the quality of requirements. The technique's ability to capture and represent requirements rigorously, but still in a form comprehensible to business people, could provide a powerful analysis tool for system developers. A case study conducted recently by the author attempted to synthesise ORM with the concepts and definitions provided by the BRG. In this way, business rules discovered in the organisation were expressed in the natural language of ORM. The results from this case study suggested that a single conceptual framework, meaningful to both business people and analysts, is a realisable goal. Exploiting the expressive simplicity of conceptual modelling techniques to articulate an organisation's business rules could help to fill a significant requirements gap.

Keywords

Business rules, conceptual modelling, requirements engineering.

1. INTRODUCTION

Although computer systems have been developed for business and government organisations for more than 40 years, recent industry surveys (The Standish Group, 1995; OASIG, 1996; Whittaker, 1999; National Audit Office, 2001; Conference Board Survey, 2001) have all indicated that most I.T. projects still end in failure. Many of these surveys and the findings of other authors (Glass, 1997; Leffingwell, 1997) have concluded that inadequately specified requirements are one the primary causes of I.T. failures.

Perhaps these findings are not so surprising since many contemporary data modelling techniques simply cannot represent many important system requirements. In particular, the constraints that apply to data structures cannot be easily represented, if at all, in most data modelling techniques (ter hofstede, Proper, & van der Weide, 1994). Consequently, these requirements are often not discovered until the latter stages of the development lifecycle, possibly causing serious scheduling and budgetary problems for projects.

The Business Rules Group (formerly known as GUIDE) has defined a conceptual model for rules that describe the structure and behaviour of organisations. The concepts and definitions within their report can guide analysts to seek appropriate information from domain experts by knowing "what is, and is not, a business rule" (GUIDE, 1995, p.1). Business rules originate as the policies adopted by organisations in response to constraints that act upon them (GUIDE, 1995). As such, business people represent domain experts and so it is from these people that analysts must elicit the organisation's business rules. Since these rules can only be validated when business people can challenge the analyst's understanding of them, it suggests that they must be expressed in a form that facilitates such dialogue.

Conceptual modelling approaches such as ORM, exploit the expressiveness of natural language within the formality of an underlying mathematical framework. This modelling technique relies heavily on natural language analysis. This allows the application model to be defined at the conceptual level where the analyst and domain experts can communicate more effectively (Halpin, 2002). The ability to verbalise elementary facts that describe an organisation, suggests that business rules could be expressed within these models. This would allow domain experts to become actively involved in their validation. By integrating conceptual modelling techniques within the business rules model, analysts may have a mechanism for capturing requirements more completely and accurately. That ability could help to fill a requirements gap and thus reduce the risk of project failure.

2. REQUIREMENTS ENGINEERING (RE) PROBLEMS

The problems associated with RE have been well documented for many years. Researchers have noted (Bell & Thayer T.A., 1976; Meyer, 1985) incomplete, ambiguous and inconsistent requirements are commonplace in industry and recognised the significance of these inadequacies on software quality. More recently, various studies (The Standish Group, 1995; OASIG, 1996; National Audit Office, 2001; Conference Board Survey, 2001) have confirmed these findings and the magnitude of the problems caused by poorly defined requirements. The Standish Group (1995) discovered that less than 20% of the 8000 American IT projects they surveyed, delivered a system that was complete, on budget and within schedule. One third of I.T. projects they surveyed were never completed. The remaining projects were only partially completed and often well over schedule and budget. Managers attributed the cause of over 40% of project failures to problems relating to the gathering, specifying and validating of requirements. These findings suggest that improving the quality of requirements can significantly reduce the risk of project failure.

RE is a complex task requiring the analyst to bridge the gap between two specialist perspectives on a system (Ryan, 1993). The domain expert views the system from a high conceptual level and it is a business-orientated perspective. The analyst must transform this view into a complete and precise specification that is necessarily computer-system orientated (Pohl, 1993). Each of these perspectives is associated with concepts and a language used to express them. This presents a significant problem since in order to agree on what the required system should do, domain experts and analysts need to effectively communicate (Pohl, 1993). Unless the domain expert can challenge the analyst's understanding of a system, one would expect such an agreement to be difficult to attain.

3. CONCEPTUAL APPROACHES TO RE USING NATURAL LANGUAGE

It has been suggested that natural language (NL) is a valuable tool for communicating with domain experts (Nijssen & Halpin, 1989; Hoppenbrouwers, Vos & Hoppenbrouwers 1996; Mazza, Fairclough, & Melton, 1994). However, NL is currently not widely used in industry for two major reasons. Firstly, NL does not lend itself for the expression of implementation concepts such as data retrieval and storage (Kristen, 1994). Secondly, NL possesses an inherent tendency towards informality and imprecision (Gamut, 1991). Despite these apparent shortcomings, a number of NL approaches to RE have been developed as a mechanism for defining a conceptual model of a system. The ORM approach is discussed below and how it attempts to overcome the two problems highlighted above are explained.

3.1 The ORM Approach

Object Role Modelling (ORM) is a well-established approach for conceptual modelling and is associated with a methodology known as Natural language for Information Analysis Method (NIAM), (Nijssen & Halpin, 1989). A central concept within ORM is the idea of objects (entities or values) playing roles (parts in relationships).

The application domain is modelled at a high conceptual level using terms familiar to domain experts in the form of elementary facts, constraints and derivation rules. Implementation issues, both logical and physical are ignored, allowing analysts to focus on the problem domain and to communicate effectively with domain experts.

Elementary sentences are constructed from discussions with domain experts that describe the roles played by objects e.g. 'Customer places Order'. In this example, there are two objects; 'Customer' and 'Order' that are associated via the role 'places'. ORM will also support the expression of any type of arity. This allows facts to be expressed in more naturally, e.g. 'Customer 123 at Time '9am' rents a Car 'ABC', represents a ternary fact type since it involves 3 objects.

ORM allows analysts to capture many more constraints, enforcing data integrity, than is possible is most other modelling approaches (Halpin, 2002). This ability allows analysts to explicitly represent these details at a conceptual level, not only facilitating validation by domain experts, but also allowing automatic code generation of these rules. Uniqueness and role constraints have a parallel in all other data modelling approaches, but ORM allows many others to be expressed, such as:

Subset – where the population of one role is a subset of another

Equality – equivalent to a subset constraint, but in both directions

Exclusion – where the populations of two roles are mutually exclusive

Ring – applied to roles played by the same object i.e. recursive relationships.

(Includes asymmetric, intransitivity and acyclicity.)

Subtype – ORM also supports multiple inheritance

These constraints are added to the model as graphical annotations to structural details previously captured. As such, ORM appears to support the expression of many of the constraints that affect the behaviour of systems. Such details are generally weakly supported, if at all, by many other modelling techniques (Halpin, 2002).

Details captured within ORM are mapped into table structures by using an algorithm incorporated into Visio Modeller, ORM's automated tool support. The algorithm transforms the conceptual model into Optimal Normal Form, which is equivalent to fifth normal form (Halpin, 2001). After mapping, the database can be generated into a number of target database formats including Oracle, SQL Server, and Access. Data integrity details, as specified by the constraints annotating the conceptual model, are also implemented in the target database as functions or database triggers.

3.3 Conclusions Regarding the ORM Approach

The approach resolves the problem of informality and imprecision normally associated with natural language (Gamut, 1991). This is achieved by adopting a restrictive grammar whose syntax and structure have a formal mathematical foundation. ORM allows analysts to create graphical models, provide sample populations and facts, or to enter Formal Object Role Modelling Language (FORML) statements directly into Visio Modeller. In this sense, precision and formality is achieved immediately, although still in a form comprehensible to domain experts, rather than being preceded by a phase where requirements are expressed informally.

The second problem associated with the use of natural language concerns its weakness for expressing implementation concepts (Kristen, 1994). In the ORM approach, the domain is modelled at a high conceptual level in a fact-finding process. Implementation concepts, both logical and physical, are simply not present within ORM models. However, Visio Modeller is still capable of generating databases from these models. Implementation issues are addressed internally, out of sight of even the analyst, by using algorithms to normalise ORM models and to implement them as relational tables and constraints (where supported by the target DBMS).

4. MODELLING BUSINESS RULES

ORM distinguishes itself from many other modelling techniques in the degree of separation from technological aspects in which the application domain is described. The business rules approach for defining organisations and their behaviour is possibly complementary to this modelling technique. The business rules approach describes the organisation in terms of 'the rules that define the structure and control the operation of an enterprise' (GUIDE, 1995, p.1), independent of any technological considerations.

A project initiated by the Business Rules Group (formerly known as GUIDE), consisting of E.F. Codd, Charles Bachman and John Zachman amongst others, had as its major objective the goal of "formalising an approach for identifying and articulating business rules", (GUIDE, 1995, p.1). They defined four major categories of rules; definitions of business terms, facts relating terms to one another (which they call structural assertions), constraints that limit the behaviour of organisations (referred to as action assertions) and derivations (calculations). As such, the business rules approach does not present any new concepts unfamiliar to system developers. However, by defining the organisation and its behaviour as a collection of atomic business rules, the approach provides a framework for describing the organisation that is meaningful to both analysts and domain experts.

Business rules themselves are declarative in nature, rather than procedural. They define states that are required or prohibited, which maybe conditional, rather than the steps taken to move from one state to another. As such, other, process-orientated techniques are required to define the sequence in which they are executed.

The group argue that many techniques are available for representing the rules that define the structure of organisations (structural assertions), for example ER and OO approaches, but very few are able to describe how the behaviour of organisations are constrained (action assertions) (GUIDE, 1995). Consequently, action assertions are often neglected, but important requirements. These details may not be discovered until an attempt is made to implement these requirements as program code (GUIDE, 1995).

Recent industry surveys, such as that conducted by the Standish Group (1995), have identified inadequately specified requirements as the cause of 40% of all I.T. project failures. These surveys do not explicitly identify constraints as the source of these missing requirements. However, if analysts do not possess the tools to formally articulate them, it seems reasonable to suggest that this inability could contribute to at least some of the problems encountered.

The inability of most contemporary modelling techniques to express constraints raises an interesting possibility. Perhaps ORM constructs could be used as the language to formally articulate the constraints (action assertions) on organisations, or indeed, any other types of business rule. Further, since this conceptual modelling technique employs natural language, it suggests that these business rules can be expressed in terms familiar to domain experts. This would allow them to become actively involved in their validation. Therefore, by synthesising these two approaches into a single conceptual framework, analysts may have a mechanism for capturing requirements more completely and with greater accuracy. That ability could help to reduce the risk of project failure. To test this hypothesis, a case study involving a small Private Training Enterprise (PTE) in central Wellington was conducted. This case study is described in the next section.

5. CASE STUDY

The rationale for choosing the case, a Private Training Enterprise (PTE), was that it seems to exhibit many of the characteristics typical of enterprises in New Zealand. It is a small to medium enterprise (SME) employing around 20 people. In New Zealand, SME's employing between 0 to 20 people account for 97% of all businesses (Statistics New Zealand, 2003). Also, like 88% of small businesses, the target organisation employs Information Technology in its day to day activities (Statistics New Zealand, 2002). The final justification for selecting this case was the difficulty the organisation has experienced with the in-house systems it has developed. For example, the PTE has expended over $80,000 NZD in the development of a Student Management System. However, after 18 months since the inception of the project, the system lacks many of the features and functionality as was originally intended. Initial interviews with staff and the owner/managers suggested that the principal cause of these failures has been due to problems relating to the specification of requirements. This finding reflects a common problem with many IT projects, as detailed in section 2.

As a consequence of this organisation's characteristics and problems, it suggests that if improvements can be made to the quality of its system requirements resulting in an application that more closely matches the needs of the organisation, there is at least a possibility that similar findings may be found in other cases. To this ends, the author, in collaboration with the PTE, developed the requirements for a small but complex component of their Student Management System using the conceptual approach advocated in this paper.

5.1 System Requirements Identified in Case Study

The requirements developed for the PTE centred on the student mentoring component of their Student Management System. The aim of this subsystem is to provide student supervisor's with a mechanism that allows them guide their students in terms of the options available to them to complete NZQA qualifications. The National Qualifications Framework (NQF) is a flexible but complex structure that allows students to 'mix and match' unit standards depending upon the qualification requirements, the electives chosen by the student and their particular strengths and interests. In order for supervisors to provide effective mentoring, they need to know what unit standards the student has already achieved and what other combinations of unit standards are required to successfully complete their qualification.

In order to achieve this goal, the existing system requires significant modification so as to allow the NQF framework to be mapped into their database. This will enable the requirements of each qualification that the PTE delivers to be recorded into its database. Once completed, the specific requirements of a qualification can then be matched against actual student achievement and the various options available for completion could then be determined and used to guide the student.

5.2 Research Methods Employed in Case Study

The author chose to apply a pluralistic approach by combining two qualitative techniques in the form of action research conducted from within an explanatory case study.

The case study drew upon qualitative data sources in the form of interviews with employees within the PTE, observations of their business procedures and the documents used in carrying out those procedures. The aim of the case study was to discover the organisation's business rules relating to the student mentoring process of their operation.

Since business rules incorporate the terminology used within the organisation, it was important that these terms be rigorously examined and their meaning standardised. The technique of triangulation was applied to facilitate this process by drawing from multiple sources of evidence to ensure that these terms were accurately defined. These definitions were then presented to management and end-users with the aim of reaching consensus as to their meaning.

Action research was used in a collaborative effort involving the target organisation's management, employees and the author. The motivation for using action research was to determine and validate the system requirements for the organisation's student mentoring process. This was achieved by expressing the business rules discovered during the case study in the formal natural language FORML of the Object Role Modelling technique.

As a final check as to validity and completeness of the requirements expressed in ORM's language, a second case study will be conducted after the implementation of the student mentoring component (anticipated to occur by July, 2004). During this case study, management and end-users within the PTE will be invited to make an assessment as to how closely this component matches their actual needs. The focus of this assessment will be on the reporting function of the student management sub-system that documents student progression to date and the various options available for completion of their qualification. As such, this reporting function relies on the requirements for the sub-system as a whole to have been fully and accurately captured in order to truly reflect student progression. Indeed, data has to be extracted from every new table that was identified during the case study, when these reports are generated. Thus, the ability of this function to produce reports that accurately reflect student progression and completion options should provide a good test of how rigorously the requirements of this sub-system have been captured. Based on the observations and comments made by management and end-users during the second case study, the author will then address the central hypothesis of this paper by assessing the efficacy of employing ORM's formal natural language to express this organisations business rules.

5.3 Preliminary Findings of Case Study

As mentioned previously, the full conclusions of this research will not be known until the completion of a second case study that will evaluate the effectiveness of articulating business rules using ORM structures. However, it can be reported that management and end-users of the PTE found the conceptual approach promoted a greater degree of collaboration in the effort to determine their requirements. They attributed this to the general improvement in the transparency with which requirements were determined and expressed. They felt much more able to challenge the perceptions that the author had acquired during the case study than they had previously, when traditional analysis techniques had been employed by others during the development of their existing system. This improvement in the collaborative aspects of the systems development effort seems to have resulted from the ability to verbalise requirements in natural language.

The transparency with which requirements were determined was further promoted by focusing on the rules employed by the organisation, rather than the data-orientated abstractions that ER diagramming and OO Class diagrams tend to promote. Business rules appeared to have much more meaning to end-users and management than the concepts incorporated within the traditional modelling tools that are used to define data requirements. Certainly, from a personal perspective, it was much easier to explain my understanding of their system using this conceptual approach, since it was possible to completely ignore implementation issues that often complicated effective communication when I worked in industry.

6. CONCLUSIONS AND FUTURE RESEARCH

This paper has highlighted that the IT industry faces a number of challenges that must be overcome if the current high level of project failure is to be addressed. Central to this problem is the inability of many popular techniques to represent important categories of system requirements and to express those requirements in a form meaningful to domain experts. It has been argued that unless business people can actively participate in the validation process, a common understanding of the system and its requirements would be difficult to achieve.

The business rules model provides a conceptual framework for understanding the structure and behaviour of systems within a paradigm meaningful to both analysts and domain experts. To apply the concepts contained within the model, this approach requires a language to accurately express business rules.

Conceptual modelling approaches such as ORM exploit the expressiveness of natural language within the formality of an underlying mathematical framework. It has been suggested that ORM can be synthesised with the concepts and definitions associated with business rules, as defined by the Business Rules Group. The resulting conceptual framework should provide a rigorous, but transparent mechanism for expressing business rules.

The preliminary findings of a recently conducted case study that aimed to express an organisation's systems requirements using this framework have been encouraging. A follow-up case study will be conducted after the implementation of the system whose requirements were determined, to fully assess the merits of this approach.

REFERENCES

Bell, T. E., & Thayer T.A. (1976). "Software Requirements: Are They Really a Problem?" Second International Conference on Software Engineering, San Francisco, pp. 61-68.

Conference Board Survey. (2001). "ERP Trends Research Report" (Report No. R-1292-01-RR). New York: ERP Trends.

Gamut, L. (1991). "Logic, Language, and Meaning". Chicago: University of Chicago Press.

Glass, R. L. (1997). "Software Runaways". New Jersey: Prentice Hall.

GUIDE Business Rules Project. (1995) "Defining Business Rules ~ What Are They Really?" Retrieved March 14th, 2003 from http://www.businessrulesgroup.org/first_paper/br01c0.htm.

Halpin, T. A. (2001). "Information modeling and relational databases". San Diego: Academic Press.

Halpin, T. A. (2002). "Join Constraints". Seventh International Workshop on Evaluation of

Modeling Methods in Systems Analysis and Design, Toronto, pp. 121-131

Hoppenbrouwers, J. J., Vos, J., & Hoppenbrouwers, S. J. (1996). "NL structures and conceptual modelling: The KISS case". In Riet, R.P. van de, Burg, J.F., Vos, A.J. van der (Ed.). Applications of Natural Language to Information Systems, pp. 197-209.

Kristen, G. (1994). "Object-Orientation: The Kiss Method: From Information Architecture to Information System". Boston: Addison-Wesley Longman.

Leffingwell, D. (1997). "Calculating the return on investment from more effective requirements management". American Programmer, 10(4), 13-16.

Mazza, C., Fairclough, J., & Melton, B. (1994). "Software Engineering Standards". New York: Prentice Hall.

Meyer, B. (1985). "On Formalism in Specifications". IEEE Software, 2(1), 6-26.

National Audit Office. (2001). "Why IT Projects Fail". UK National Audit Review of IT Projects. Retrieved April 25th, 2003 from http://www.nao.gov.uk/intosai/edp/pas/010105ukpaperv2.1.pdf.

Nijssen, G. M., & Halpin, T. A. (1989). "Conceptual Schema and Relational Database Design: A fact oriented approach". Sydney: Prentice-Hall.

OASIG. (1996). "The performance of Information Technology and the role of human and organisational factors". Sheffield: Institute of Work Psychology, Sheffield University.

Pohl, K. (1993). "The Three Dimensions of Requirements Engineering". Fifth International Conference of Advanced Information Systems Engineering, Paris, pp. 275-292.

Ryan, K. (1993). "The Role of Natural Language in Requirements Engineering". IEEE International Symposium on Requirements Engineering, San Diego, pp. 240-242.

Statistics New Zealand (2002). "Information Technology Use in New Zealand 2001". Accessed February 13th, 2004. http://www.stats.govt.nz

Statistics New Zealand (2003). "Business Statistics Tables". Accessed February 13th, 2004. http://www.stats.govt.nz

ter Hofstede, A. H., Proper, A. H., & van der Weide, Th. P. (1994). "A conceptual language for the description and manipulation of complex information models". Seventeenth Annual Computer Science Conference, Australia, 16, 157-167.

The Standish Group. (1995). "The CHAOS Report". Dennis, MA: The Standish Group International.

Whittaker, B. (1999). "What Went Wrong? Unsuccessful Information Technology Projects". Information Management & Computer Security, 7(1), 23-30.