Teaching a New Dog Old Tricks: Modelling Electronic Commerce with Business Rules

Michael N Johnstone
Donald C McDermid
John R Venable

School of Information Systems
Curtin University of Technology
Perth, Western Australia
Email: {JohnstoneM, McDermidD, VenableJ}@cbs.curtin.edu.au

Abstract

This paper describes an action research study whose focus was on the suitability of a "conventional" approach for the requirements elicitation for an Electronic Commerce (EC) system. While it is acknowledged that new business models are required for EC, a basic question arises as to what extent existing requirements engineering methods are sufficient for developing EC systems. The study showed that it was possible to develop a specification for a business-to-consumer system using a "conventional" requirements engineering method, the Business Rules Diagram method, although some tailoring of the use of the constructs in the method took place.

Keywords

Information Requirements Determination; IS Research Issues; Action Research.

ELECTRONIC COMMERCE SYSTEMS DEVELOPMENT

Over the past few years, the World-Wide Web (WWW) has become a pervasive medium for communication, in both business-to-business (B2B) and business-to-consumer (B2C) electronic commerce (EC). It is so pervasive, in fact, that the Chairman of Intel, Andy Grove, has been quoted as saying that "in five years time, all companies will be Internet companies or they won't be companies at all" (Belli, 1998).

Various authors have suggested that EC systems (particularly web-based systems) differ significantly from more "traditional" systems and that new and different development methods are needed for EC projects. Bichler and Nusser (-) suggest that the development of large-scale WWW applications usually involves a large team of individuals with different skill sets (e.g. programmers, authors and graphic designers) and (paraphrasing Garzotto et al., 1993) that web design involves the "structuring of a complex information domain...making it clear and accessible to users". However, "a large team composed of members with different skills" could describe any modern large scale development. Furthermore, many EC systems are relatively small systems developed by relatively small groups for relatively small organisations. It can also be argued that the "structuring of a complex information domain...making it clear and accessible to users" applies to requirements engineering for many information systems, not just Web-based ones.

Other authors (e.g. Fruhling and Digman, 2000; Burdick et al., 1999) argue that EC systems are different due to, amongst other factors, the capacity for value-adding, differentiation between firms and the potential market size. These factors describe some aspects of the strategic nature of some EC systems, but they also apply equally to other kinds of strategic IS applications. Other authors have proposed that web-based EC applications require different development methods, for example SHDT (Bichler and Nusser, -), HDM (Garzotto et al., 1993), RMM (Isakowitz et al., 1995), OOHDM (Schwabe and Rossi, 1995), extensions to UML (Conallen, 1999), and the Scenario-based approach of Lee et al. (1999). However, a different domain (EC) is not necessarily the same as a different type of system; many EC systems involve ordinary data gathering, storage and processing for which existing methods may be appropriate. Furthermore, all of these new web-based EC development methods focus on the design of EC systems and not on requirements elicitation. Indeed, there is a dearth of literature regarding requirements elicitation in EC.

Given the above concerns about whether the (apparently) received view that EC systems are sufficiently different enough to require different development methods, this paper considers whether it is suitable to use an existing requirements elicitation method based on modelling business rules in an EC development project. In particular, the action research study described in this paper set out to establish whether the constructs found useful for modelling “traditional” information systems could be applied successfully to the domain of electronic publishing.

BUSINESS RULES IN REQUIREMENTS ENGINEERING

Despite their utility, business rules are somewhat neglected in the requirements engineering literature. One explanation is that there is some disagreement as to what actually constitutes a business rule (Loosley 1992). According to Sandy (1996), the range of definitions includes such varied things as critical success factors, quality goals and database integrity constraints. We consider it is important that the definition of a business rule stipulates something of the context and nature of a business rule as well as identifying its constructs. McDermid (1998, p20) defines a business rule as *‘…an explicit state change context in an organisation which describes the states, conditions and signals associated with events that either change the state of a human activity system so that subsequently it will respond differently to external stimuli or reinforce the constraints which govern a human activity system’*, a definition we shall use from here onwards.

THE BUSINESS RULES DIAGRAM METHOD

A recent approach used to capture business rules is that of McDermid (1998). This approach, called the Business Rules Diagram (BRD) method, uses a state-based model notation similar to flowcharts. As an approach, the BRD method is positioned between the use case approach of Jacobson *et al.* (1992) and more complex object models. A business rule, as defined by McDermid (1998), contains four explicit constructs, these being states, events, conditions and signals (figure 1). Connected combinations of these constructs make up a User Business Rule Diagram (UBRD). One other construct is the Harel blob (Harel 1988), which encapsulates other constructs and is used to model selection or simultaneous action. The use of the blob construct in the full BRD distinguishes the BRD from the simpler UBRD. States reflect the status of a system or one of its components. For example, a visitor to an electronic journal web site might traverse the states *visiting, subscribed* and *unsubscribed*. Events are actions carried out internally by the organisation. An important role of the event is to avoid specifying processing detail which is kept separate from policy-level business rules as will be discussed later. Conditions define the criteria by which objects of interest in the business move from one state to the next as events take place and are sometimes known as "if-then rules" in other systems. Lastly, signals either enter or leave the human activity system. Signals that enter the system typically initiate activity within the system and are called triggers. Signals which leave the system serve to inform those outside the system boundary about what has occurred inside the system and are called messages.

Business rules are split into policy, processing and implementation-level rules. Policy rules are the most general and are represented by externally verifiable state to state transitions. An example of a policy rule is: "any visitor to the web site may become a subscriber by providing identification and paying the appropriate fee". Typically, the states here would be *unsubscribed* and *subscribed* with the intervening event being "subscribe to journal". Here, the states are externally verifiable as the visitor, who is external to the system, requests a subscription (a trigger) and is sent a confirmation of the subscription by email (a message).

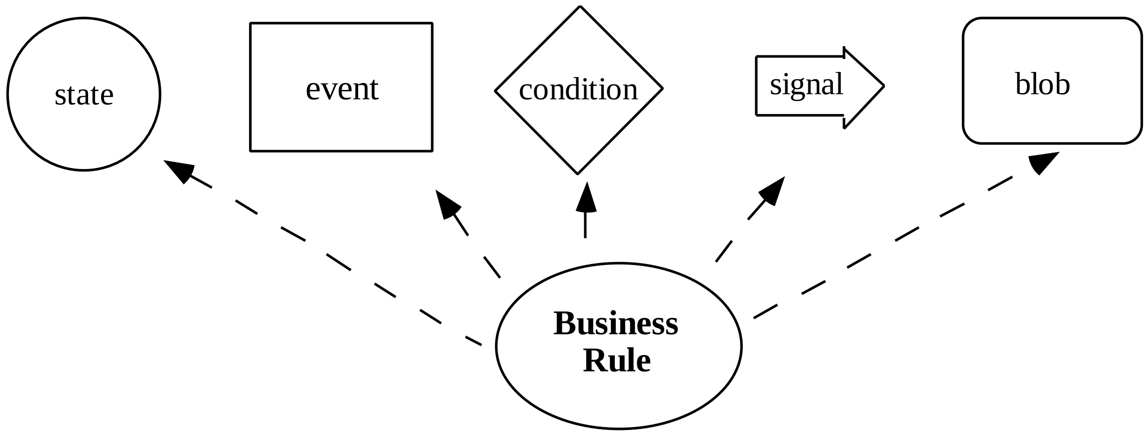

Figure 1: Graphic Notation used in the Business Rule Diagram

The next level, processing rules, are similar to policy rules in that they follow the same state-event-state pattern but they may not necessarily have externally verifiable states and thus may be considerably more complex. An example of a processing rule is: "only if the subscriber’s name, credit card number and expiry date are correctly entered will the subscriber be accepted as a valid subscriber". The lowest level, implementation rules, describe the system in detail and are beyond the scope of this study. To date, studies have been carried out concerning both policy rules (McDermid, 1998) and processing rules (Johnstone and McDermid, 1999). This study builds on earlier work by testing several concepts suggested by the latter. What makes this study different from prior studies is that the BRD method is used here to model a completely new situation (a "greenfield" site) as distinct from modelling an existing system.

RESEARCH METHODOLOGY

Action research was the underlying research approach adopted in this study. Action research has been used in a wide variety of organisational and problem-solving situations. For example, see Lau (1997) for a range of information systems applications of action research. There is also a long-standing tradition of action research in the development and suitability of IS development methods in organisational situations, including work by Wood-Harper et al. (1985), more recently Johnstone and McDermid (1999) and Mathiassen (1998, p102) who notes that "*Action research brings relevance to the research process while supplementary approaches improve the validity and reliability of the research results*.".

The specific research framework used to guide this study was that of Venable and Travis (1999), which extended Nunamaker et al. (1991) to include both action research as a method of *in situ investigation* and the creation and refinement of systems development methods (e.g. in this case the BRD for developing EC systems) as a suitable topic for systems development in IS research (see figure 2). The model in figure 2 is not dissimilar to Lewin's cyclic model of action research (Burns, 1990) or to Kolb's experiential learning cycle (Kolb et al., 1979) as all three models have a common theme of a reflection/action learning cycle. In the case of this research, the action learning cycle encompassed the *theory building*, *system development* (of the BRD and how it should be used), and *in situ investigation* (using action research) of figure 2.

A body of work now serves as guidelines for researchers in the practice of action research as well as in evaluating the results of action research studies (Baskerville and Wood-Harper (1996a, 1996b), Baskerville and Pries-Heje (1999). The guidelines for conducting interpretive research suggested by Klein and Myers (1999) were followed in this study. For example, a client-systems infrastructure was agreed (Baskerville and Wood-Harper, 1996b), many formal cycles of action/reflection took place (Burns, 1990) and so on.

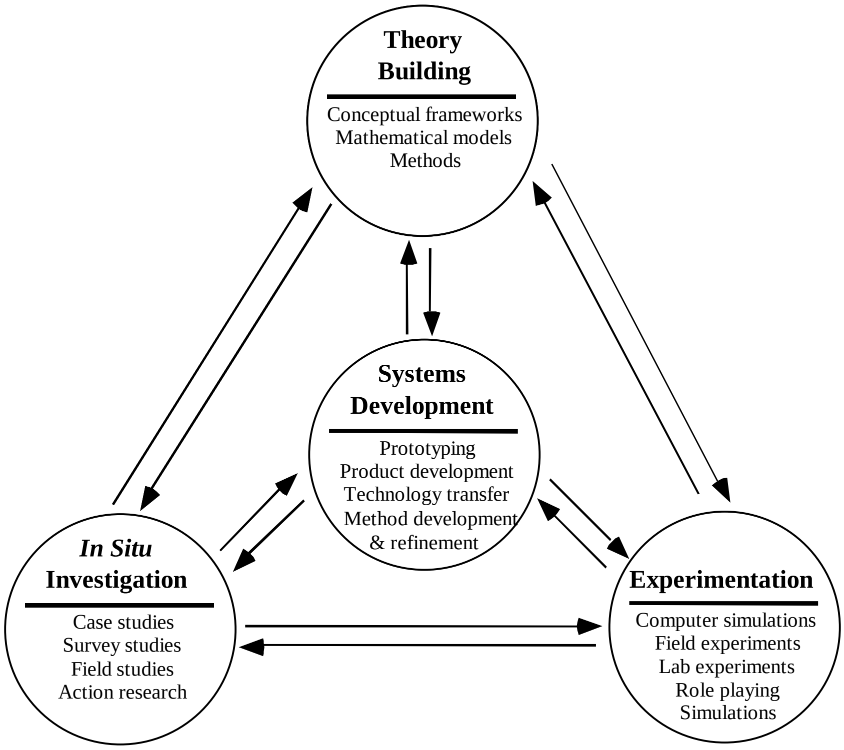

Figure 2: Systems Development in IS Research

(Adapted from Venable and Travis, 1999 after Nunamaker et al., 1991).

THE ACTION RESEARCH STUDY

In this section, we will describe the environment and scope of the study as well as the progress made by the team at each stage of the development. The client was the Management School of a large Australian university. The participants in the study were a systems analyst with 15 years experience in information systems development (the researcher) and two users (henceforth referred to as client representatives or clients) with experience in paper journal publishing and web-oriented peer review systems. The client-system infrastructure was notable for its particularly rapid learning cycle (often, only a week between cycles). As suggested by Baskerville and Wood-Harper (1996b), copious records were kept during each cycle including field notes and minutes of meetings as well as taped post-project semi-structured interviews. The clients agreed to be participants in the study and to help in the research. The clients aided the research by discussing the suitability of the BRD method and notation and suggesting improvements. This was done during the action research study itself in post-intervention interviews.

The problem domain was that of electronic publishing. The intention was to publish two issues of an electronic journal per year, the journal to be made available on the WWW with on-line subscription using credit cards an important requirement. The journal abstracts were to be open to a casual visitor to the web site with the full text of articles to be available only to paid-up subscribers. This problem domain is clearly toward the smaller end of B2C EC systems, but still exemplifies the need for the use of business rules.

In order to ensure that the participants had sufficient and appropriate knowledge at each step of the process, clients were trained in the BRD method. Each workshop session contained a tutorial element where the researcher discussed the concepts of the BRD method with the clients and provided worked examples via case study material to reinforce the learning opportunities provided by the tutorials.

The process used in the method to create a complete BRD is defined by McDermid (1998) as:

- identify candidate business rules;
- identify candidate events and signals;
- identify candidate objects and construct Object Life Histories (OLHs);
- construct User Business Rule Diagrams (UBRDs);
- construct Business Rules Diagram; and
- construct Event Specification Table (EST).*

* The last step of the method is beyond the scope of this paper.

Step 1: Identification of Candidate Business Rules

Initially, the team generated 35 business rules (which later grew to 84) covering many aspects of the publication of a journal whether paper or electronic. At this stage the rules were not classified as policy, processing or implementation rules although it was noted that only one of the rules "brainstormed" actually mapped to the state-event-state meta rule established for policy-level business rules. Many of the rules appeared to be concerned with access to the web site which was to support the journal e.g. "only subscribers may access the full text of journal articles" and "any visitor may browse the journal abstracts". Despite the focus on high-level rules during the tutorial, both clients appeared overly concerned with implementation details such as how the journal articles could be protected from unauthorised copying using PDF files.

Step 2: Identification of Candidate Events and Signals

Both clients appeared comfortable with the concept and application of step two of the BRD (gauged from response to the tutorial) and were comfortable with the idea of modelling the paper submission process as a means of generating events and signals. This is rather like a use case dialogue, but potentially from several actor viewpoints.

A sequence of events was mapped out concerning submission which clearly showed objects (author, editor, paper) traversing state-to-state transitions via events (e.g. paper is written, submitted, accepted, rejected etc.). It was interesting to note that the event sequence (which in many respects resembled a use case dialogue) was just as valid for a paper publication as for an electronic journal. This meant that the model was probably at the right level of abstraction as it did not contain any implementation details.

The clients had no problems in categorising the statements of the use case dialogue into events, messages and triggers. This indicated that they had a clear understanding of the system boundary and what was internal (events) and external (signals).

Step 3: Identification of Candidate Objects and Object Life Histories

During this step the clients re-evaluated, stripped and reconstructed the model based on an insight to do with differing views of the purpose of the web site being designed to support the journal. The problem of the rules not being true business rules (according to the given meta-rule) was discussed at length. It was clear that one client was focussing entirely on the web site while the other was taking a more holistic approach to the development of the system i.e. one client was happy to consider publication within the framework of the system but the other client felt that this had nothing to do with the web site and thus was out of scope. These different viewpoints explain the difference in the business rules generated thus far as clearly many of the rules were "access rules" for the web site.

The team elected to focus on the web site rules but decided not to discard the work done in the last step concerning publication in case it was needed later. One client was particularly concerned that important information might be lost by taking too narrow a view of the system.

The team categorised the business rules into two classes, web site rules and others (including publication rules). During the categorisation process one client noticed that one rule (Subscribers will be provided with access codes) was a sub-rule of another (Only Subscribers can access journals). This would indicate that the clients did understand the concept of linking processing rules to parent policy rules.

Following the tutorial session (which covered objects and object life histories), the clients had no difficulty in identifying a "visitor" object and modelling a life history for it, although they did not refer back to the rule list to find the states of this object (really a subscriber to the journal). Similarly, the clients also identified an "article" object and modelled a history for it in the same way. It was interesting that despite the previous discussion about separating publication from the web site, the clients had no conceptual or practical difficulty in working with the article object, even though this was a major object from the last step which focussed on publication.

Step 4: Construction of User Business Rules Diagrams

The team derived a dozen use cases from the existing business rule set and then, for each use case, role-played a typical dialogue between the system and a user from which the first-cut UBRD was developed *in-situ*. The team then considered behaviour that was atypical (e.g. a visitor to the site attempts to subscribe but the amount required for the transaction exceeds their credit card limit) and melded this behaviour into the first-cut UBRD.

In this step the apparent simplicity of the BRD method was particularly advantageous as the clients were able to lead the discussion about the validity of the proposed behaviour (i.e. does the web site as modelled reflect the business rules?) and to verify assertions by examining and modifying the UBRD as necessary.

By using the diagram to reason about the problem, the clients were able to specify constraints (in the form of partial business rules or conditions) that were not apparent using a textual representation (i.e. a list of business rules or events and signals). Figure 3 shows the Process Rules Diagram generated from one use case. In figure 3, the dashed circles indicate sub-states which act as placeholders for internal events and represent re-entrant points where a system can "roll-back" in the case of a transaction (event) not being completed. Since this diagram is modelling events at the processing level it is focussed on sequencing. The equivalent policy level BRD is as simple as the states "unsubscribed" and "subscribed", separated by the event "subscribe to journal". In this case the "subscribe to journal" event is composed of the model expressed in figure 4 (introduced in the next section), which is a processing rules diagram containing sub-states that are not necessarily externally verifiable.

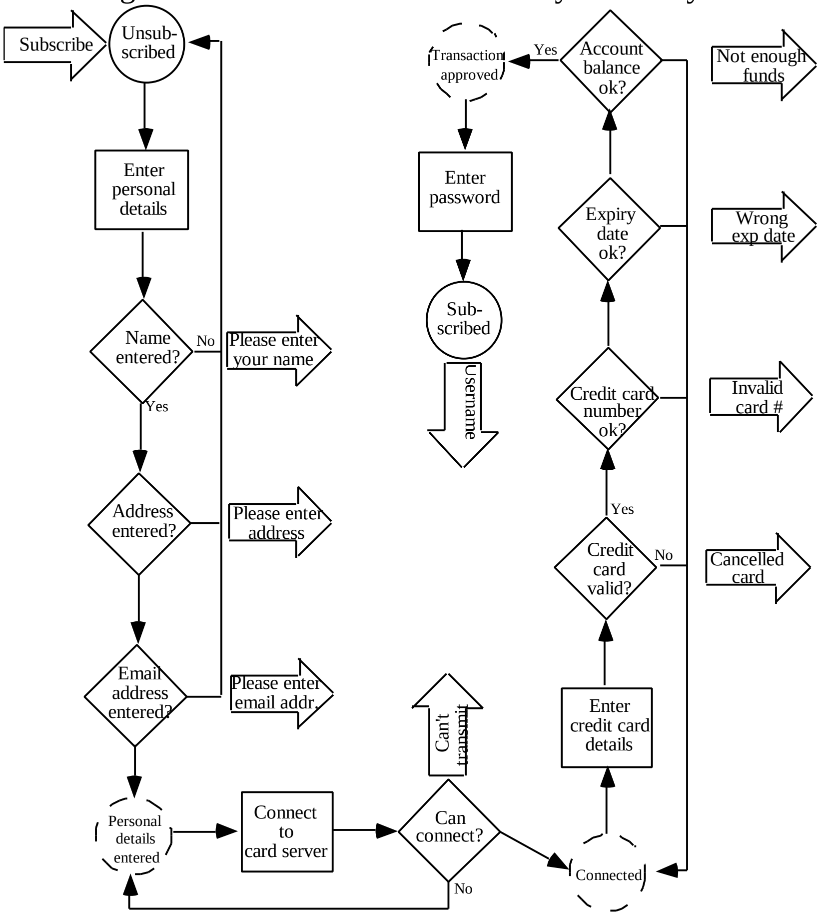

Figure 3: A Processing-Level UBRD Covering the Use Case "Subscribe to Journal"
Note: The circles with dashed lines represent substates.

As more UBRDs were developed (and thus the clients became more familiar with the method) the clients omitted the use case dialogue step and chose to draw the processing-level UBRD directly. The clients also began to model abnormal behaviour directly with the UBRD. At this stage the clients were able to take full control of the diagram and used it to reason about the logic of web site navigation. This is really a design issue but highlights the fact that, at the processing level, the BRD method is much closer to design than when at the policy level. The clients also used the UBRD to analyse the expected interactions between a user and the system as well as using it to check the logic and validity of the business rules themselves.

LESSONS LEARNED FROM THE STUDY

Modelling Concepts Unique to a Web-based System

This question arose when the team attempted to model the common "click here to return to home page" event. This can be achieved in a PRD in several ways. Firstly, by linking any set of substates to a common event ("return to home page") and then a corresponding state ("at home page"). Alternatively, the entire PRD can be encapsulated in a Harel blob which is then linked to an event and another blob containing as many states as there are links to other web pages. How this type of non-procedural behaviour can be modelled with a PRD is shown in figure 4. Perhaps the most striking difference between the UBRD and the PRD is the lack of procedural information in the latter. As can be seen from figure 4, the encapsulating properties of the blob construct account for the ability of the PRD to model a heterogeneous network structure such as a web site. The process information is not lost, however, but stored in a different representation (the event specification table or EST) which is used in conjunction with the PRD to reason about the business rules of a system.

Attempting to model non-procedural behaviour is not recommended for the UBRD at the processing level, however, as it would make the diagram less clear (given that each substate which allows the transition would need a link back to this "at home page" state). Further, there may be business rules where this type of behaviour is not appropriate (e.g. in the middle of a credit card transaction). This issue could be dealt with by the use of guard conditions that specify that the "return to home page" event can only occur *if and only if* the guard is not true.

Client Take-up of the Method

The BRD method has been tested across a limited number of areas (Portbilling, Student registration and Interior design) and while the results are promising, it needs to be evaluated more rigorously in different problem domains. A prior study (Johnstone and McDermid, 1999) used a half-day training course to give end-users the knowledge required to use the BRD method.

A problem identified with this approach is that it requires a user to assimilate a great deal of information concerning possibly unfamiliar concepts before attempting to use the method. In contrast, the training component of this study was structured quite differently in that each workshop session contained a tutorial element where the researcher conducted a presentation, discussed the concepts of the BRD method covered in the tutorial with the users and provided worked examples via case study material to reinforce the learning opportunities provided by the tutorials. Each presentation contained clear goals, a review of the previous session and worked examples in addition to the main topic.

The users in this study expressed that this approach made the concepts easily understandable. By the fifth session (the first without a tutorial element), the users were confident enough to take control of the UBRD being developed and use it to reason about the issues of concern to them. This confidence in the use of the method was reflected in the way that the users skipped the use case dialogue stage when they felt it was impeding their progress (without losing any details or business rules required in the UBRD). Further, by using the UBRD they were able to propose and test new business rules that only emerged at this stage. This action research study has provided further confirmation that users can relate to and use the BRD method as a reasoning tool to develop requirements specifications.

Role of Substates in the PRD

The idea that substates could exist (in the BRD) was raised by McDermid (1998) but this study was the first to use them in a BRD method requirements model. The particular role that substates held in the model was that of internal placeholders for re-entrancy or roll-back, a common design construct. This would imply that the PRD (as distinct from the BRD) is more closely linked to the design phase of the system development life cycle, a not unexpected result.

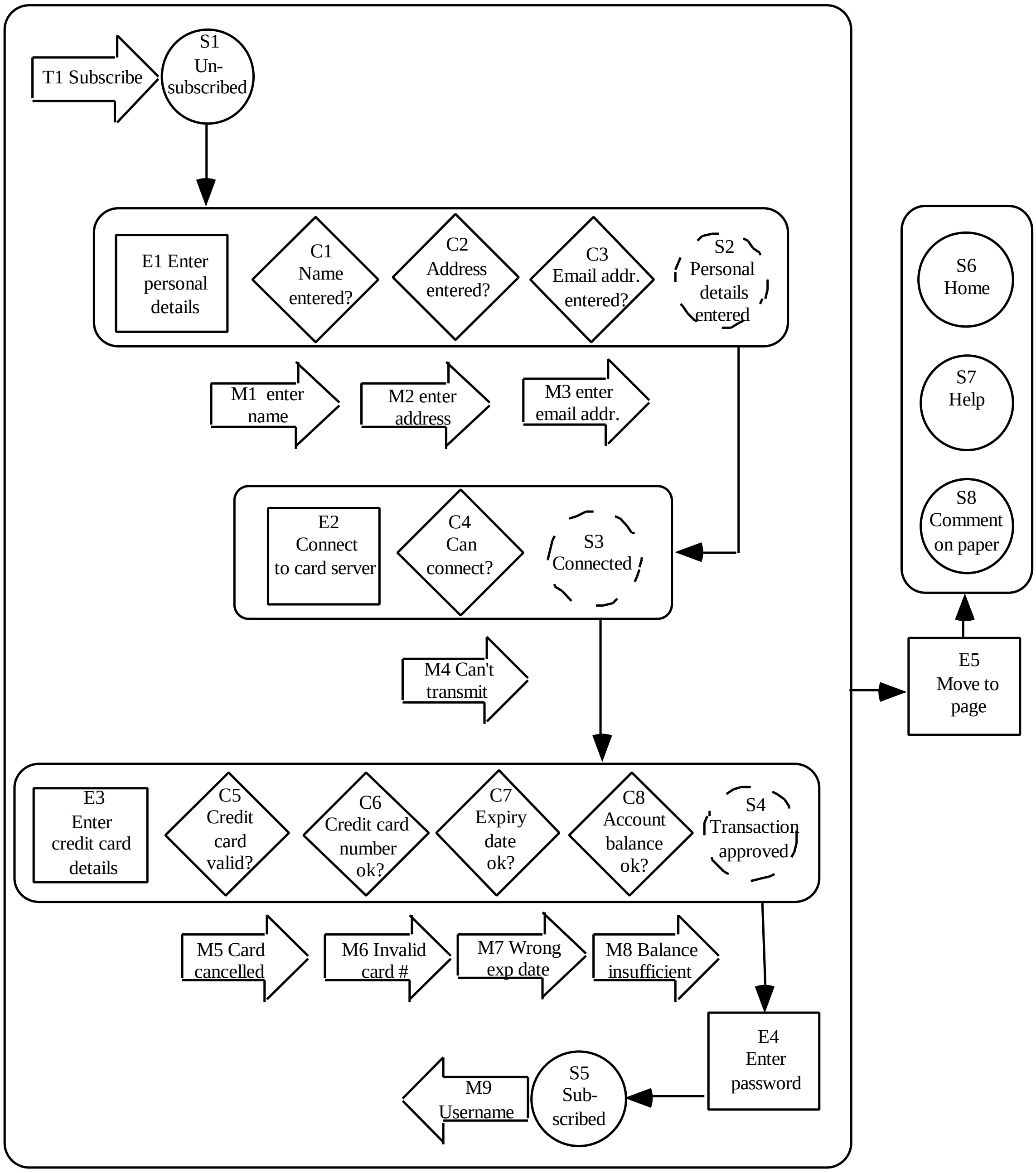

Figure 4: A Process Rules Diagram (PRD) Covering the Use Case "Subscribe to Journal"

Returning to figure 4, each substate is encapsulated by a blob. This provides for the notion of re-entrancy to part of a PRD which can be likened to reuse in implementation or determining responsibilities in object-oriented design. Consider the following situation: The business rules of the system allow for one interactive paper per issue. This paper is published and selected visitors to the web site may be invited to comment on the paper interactively, but these visitors may not necessarily be subscribers to the journal. Some basic information about these "live commentators" is required, similar to that needed for subscribers. Using substates, a modeller can effectively "call" the required sub-module of the "subscribe to journal" PRD (in figure 4, the blob containing event E1 and substate S2) without having to duplicate this functionality in another PRD.

Transition from the BRD to the PRD

It has been suggested in earlier work (Johnstone and McDermid, 1999) that one way to handle complexity in the method would be to model abstraction by using the event construct to link a BRD with a corresponding PRD. An example of this linkage can be clearly seen in the policy business rule "Any visitor to the web site may become a subscriber by providing identification and paying the appropriate fee". The states are *unsubscribed* and *subscribed* with the event being *subscribe to journal*. The processing-level detail of *subscribe to journal* is what has been modelled in figures 3 and 4. The results from this study appear to indicate that linking the different levels of diagrams using events is a viable way to manage different levels of abstraction.

In the context of this study, action research was used as both a problem-solving approach to the organisational problem of determining the requirements for an EC application and a research approach to confirming that the BRD is able to model an EC application. A second aim of the study was to examine, in an action research situation, the effectiveness of the BRD notation for representing processing rules via a Processing Rules Diagram or PRD (this is both *theory building* and *system development* in figure 2).

The research question, which involves examining levels of abstract representation within the BRD method, has, as its basic premise, the notion that the notation of the BRD can be used to build a less abstract business model (a PRD). Further, it assumes that the current structure of policy, processing and implementation rules is a viable taxonomy. Clearly, these assumptions are the parameters of the *theory building* stage and, coupled with a technique that allows the business rules to be easily separated into their respective classes as well as a method for linking the two levels of diagram, provide guidance for the *in situ investigation* stage of action research.

Given that McDermid (1998) has defined an ordered set of steps required to produce a BRD, these steps (up to and including the development of UBRDs) were followed in an action research situation with a group of clients where the clients were taught the notation and stages of the BRD method and then attempted to model a problem situation, aided by a trained business analyst acting as the group facilitator. This is represented by the *in situ investigation* stage in figure 2. The material gathered during the study provided data for observation/reflection (returning to the *theory building* stage) which then acted as a trigger for modifying/improving the BRD method (*system development* in figure 2).

BRD Method Meta-rules

A meta-rule defined and used in prior studies (McDermid, 1998; Johnstone and McDermid, 1999), that business rules occur as state-event-state transitions appears to hold even at the processing level (using substates rather than externally verifiable states). This is quite an important result as to date there has been no evidence that such an assertion would be true. Further work would need to be carried out to examine the effect of breaking this meta-rule at the processing level. This does give rise to speculation about whether this meta-rule would either hold true or even be useful at a still lower level of abstraction, that of implementation rules.

SUMMARY

This work set out to research the BRD method in the context of an EC problem situation. Specifically, it explored the effectiveness of the policy-level BRD notation for representing processing rules via a Processing Rules Diagram or PRD. The following was uncovered in this study.

With regard to requirements engineering in EC systems, the BRD method (in particular, the BRD/PRD combination) was considered to be a useful modelling tool in a Web environment as both policy (what is required) and processing (how is it done) were captured with simple but powerful modelling concepts. While the EC system analysed was toward the small end of EC systems, in our opinion the analysis and modelling of business rules would be all the more important for larger EC systems with more requirements and more complex architectures.

One of the most exciting aspects of this study was how action research facilitated user-led requirements determination. This was particularly evident when constructing the later UBRDs as the clients swiftly became confident enough to take control of the modelling situation with the facilitating analyst in a secondary role. This would strongly suggest that the BRD method has a balance of simplicity and expressive power that makes it an effective requirements elicitation tool.

This study explored some of the important meta-rules of the BRD method (e.g. state-event-state transitions) and proposed new ones based on the experiences gained in the action research study. In particular, that the current BRD method notation and semantics is adequate for capturing and modelling processing-level business rules.

REFERENCES

Baskerville, R. and Pries-Heje, J. (1999). 'Grounded action research: a method for understanding IT in practice'. Accounting, Management and Information Technology, (9), pp1-23.

Baskerville, R. and Wood-Harper, T. (1996a). 'A Taxonomy of Action Research Methods'. *Working Paper PAP012-0.05,* Institute for Informatics and Management Accounting, Copenhagen Business School.

Baskerville, R. L., and Wood-Harper, A. T. (1996b). 'A critical perspective on action research as a method for information systems research.' *Journal of Information Technology* 11, pp235-246.

Belli, J. (1998). 'ERP: Is It Now Electronic Requirements Planning?'. MidRange ERP Magazine, Sep.

Bichler, M., and Nusser, S. (-). 'Modular Design of Complex Web-Applications with SHDT'. URL http://www.wicky.wu-wein.ac.at/w3dt/wetice/wetice.html.

Burdick, D., Bond, B., Miklovic, D., Pond, K., and Eschinger, C. (1999). 'C-Commerce: The New Arena for Business Applications'. Gartner Advisory Report MRR-09011999-15, Gartner Group.

Burns, R. (1990). *Introduction to Research Methods in Education*. Melbourne: Longman Cheshire.

Conallen, J. (1999). 'Modeling Web Application Architectures with UML'. *Communications of the ACM, 42*(10), pp63-70.

Fruhling, A., and Digman, L. A. (2000). 'The Impact of Electronic Commerce on Business-Level Strategies'. *Journal of Electronic Commerce Research, 1*(1).

Garzotto, F., Paolini, P., and Schwabe, D. (1993). 'HDM - A Model-based Approach to Hypertext Application Design'. *ACM Transactions on Information Systems, 11*(1), pp1-26.

Harel, D. (1988). 'On Visual Formalisms'. *Communications of the ACM, 31*(5), pp514-30.

Isakowitz, T., Stohr, E. A., and Balasubranamian, P. (1995). 'RMM: A Methodology for Structured Hypermedia Design'. *Communications of the ACM, 38*(8), pp34-44.

Jacobson, I., Christerson, M., Jonsson, P., and Övergaard. G. (1992). *Object-Oriented Software Engineering*. Reading, MA: Addison-Wesley.

Johnstone, M. N., and McDermid, D. C. (1999). 'Extending and Validating the Business Rules Diagram Method'. *Pages 449-461 of*: Hope, B. and Yoong, P. (eds), *Proceedings of the 10th Australasian Conference on Information Systems*, Wellington, New Zealand.

Klein, H. K. and Myers, M. D. (1999) A Set of Principles for Conducting and Evaluating Interpretive Field Studies in Information Systems, MIS Quarterly *23*(1), March, pp. 67-93.

Kolb, D. A., Rubin, I. M., and McIntyre, J. M. (1979). *Organizational Psychology: An Experiential Approach*. Englewood Cliffs, New Jersey: Prentice-Hall.

Lau, F. (1997). 'A Review of the Use of Action Research in Information Systems Studies' in A. S. Lee, Liebenau, J. and DeGross, J. I. (eds*) Proceedings of the IFIP TC8 WG 8.2 International Conference on Information Systems and Qualitative Research*, (pp31-68). London: Chapman and Hall.

Lee, H., Lee, C., and Yoo, C. (1999). 'A Scenario-based Object-oriented Hypermedia Design Methodology'. *Information & Management, 36*, pp121-38.

Loosley, C. (1992). 'Separation and Integration in the Zachman Framework'. *Data Base Newsletter, 20*(1).

Mathiassen, L. (1998) Reflective Systems Development, *Scandinavian Journal of Information Systems 10*(1), pp. 67-117.

McDermid, D. C. (1998). 'The Development of the Business Rules Diagram', PhD thesis, Curtin University of Technology.

Nunamaker, J. F. Jr., Chen, M. and Purdin, T. D. M. (1991). 'Systems Development in Information Systems Research'. *Journal of Management Information Systems, 7(3)*, pp89-106.

Sandy, G. (1996). 'The Nature and Structuring of Organisational Rules'. Proceedings of the 7th Australasian Conference on Information Systems, Hobart, Australia.

Schwabe, D., and Rossi, G. (1995). 'The Object-Oriented Hypermedia Design Model'. *Communications of the ACM, 38(8)*, pp45-46.

Venable, J. R., and Travis, J. (1999). 'Using a Group Support System for the Distributed Application of Soft Systems Methodology'. *Pages 1105-1117 of*: Hope, B. and Yoong, P. (eds), *Proceedings of the 10th Australasian Conference on Information Systems*, Wellington, New Zealand.

Wood-Harper, A. T., Antill, L., and Avison. D. E. (1985). *Information Systems Definition: The Multiview Approach*. Oxford: Blackwell.

COPYRIGHT