Extending the Business Rules Diagram Method to Web Based System Design

M. N. Johnstone, D. C. McDermid
J R Venable

1 School of Computing and Information Science
Edith Cowan University
e-mail: {m.johnstone, d.mcdermid}@ecu.edu.au

2 School of Information Systems
Curtin University of Technology
e-mail: jvenable@cbs.curtin.edu.au

Abstract
This paper describes two action research studies that focused on evaluating the use of an information system development (ISD) method that was extended for the development of web-based systems that support business-to-consumer electronic commerce (B2C EC). A conventional ISD method (the Business Rules Diagram or BRD method) was extended to support the design of processing and navigation for web-based systems. The paper describes the rationale and the extensions. The two studies showed that the participating development teams were able to successfully use the extended BRD method to elicit, analyse, and specify the detailed processing and navigation for two web-based B2C EC systems.

Keywords
Information Requirements Determination; Design; Web Development; Action Research.

INTRODUCTION
The business of electronic commerce (EC) and the provision of the web-based systems that enable it are still new and novel in many respects. New business models have been developed and different forms of web-based systems, e.g. business-to-consumer (B2C) and business-to-business (B2B) systems have been developed. There are new objectives in achieving business benefits, such as reduction of costs through disintermediation or the provision of new kinds of services. The systems area has progressed from the advent of web sites to the provision of web applications. New web-enabled technologies upon which to base systems have been developed, such as XML, Java, scripting languages, and servlets. New areas, emphasising the design of systems have been developed, such as navigation by the user of a web site or application and multi-tiered architectures. In many cases, the users are outside of the organisation and traditional means of requirements elicitation become more difficult or impossible. To address these and other novel aspects, a range of web application design methods have been developed, for example SHDT (Bichler and Nusser, n.d.), HDM (Garzotto et al., 1993), RMM (Isakowitz et al., 1995), OOHDM (Schwabe and Rossi, 1995) and the Scenario-based approach of Lee et al. (1999). Development of web-based EC systems is an area receiving a considerable amount of research attention.

Nevertheless, as much as things change, some things remain the same, including some of the fundamental needs of IS development (ISD). There is still a need to analyse requirements for a system, even an EC system. While there is considerable evidence (Fruhling and Digman, 2000; Burdick et al., 1999) that the "business rules" of EC systems are different due to factors such as the capacity for value-adding, differentiation between firms and the potential market size, business rules about the circumstances and ways in which processing may or must be performed still need to be addressed and analysed during system development. However, all of the methods named in the previous paragraph focus on the design of EC systems and not on requirements elicitation and analysis.

Furthermore, the methods above emphasise navigation, but provide scant support for design of system processing. For simple web sites, the hyperlink and the web page were appropriate primary elements to model. However, as EC systems progress from static web sites through dynamic web sites to full-blown web-based applications, their processing requirements are increasingly complex and important. The designers of today’s web applications need to design diverse processing elements, such as scripts, servlets, calls to APIs etc. Designers of web applications need a development method with appropriate design artifacts to model both navigation and processing.
System developers need effective means both for eliciting and analysing the business rules for today’s complex web-based systems and for designing their navigation and processing. This paper proposes extensions to McDermaid’s (1998) business rules diagram (BRD) method that support both analysis and design of web-based EC systems. The BRD method was chosen over other techniques (such as those found in UML), because it is considered a) to encapsulate the critical constructs necessary for requirements specification within a single diagram and b) to offer the opportunity to seamlessly develop the specification through to design, again using the same basic constructs (McDermid, 1998). The domain of web-based EC systems was chosen because, while such domains share many common specification and design issues with traditional systems, they also present some unique challenges and so robustness of this technique in terms of applicability to different kinds of domain was also being tested. The extended method was evaluated using two action research studies. The paper provides some evidence that, at least in the domains studied, it is possible to extend and use an existing ISD method for use in EC system development projects, to successfully elicit and analyse both detailed processing and navigation requirements, and further, to articulate successfully from analysis to design.

WEB-BASED SYSTEM DEVELOPMENT METHODS

It is instructive to examine the more common web design methods as mentioned above, i.e. SHDT (Bichler and Nusser, n.d.), HDM (Garzotto et al., 1993), RMM (Isakowitz et al., 1995), OOHDM (Schwabe and Rossi, 1995) and the Scenario-based approach of Lee et al. (1999).

The Relationship Management Methodology (RMM) of Isakowitz et al. (1995) adapts the constructs of an E-R model to the elements of a web site. In RMM, data modelling follows similar rules to that of E-R modelling with the addition of the concept of “slices” (modeller-defined attribute groupings) to manage complexity in the model. The navigation elements include within-entity links (both uni and bi-directional), groupings (a menu-like mechanism for facilitating access to other parts of a system), indices (e.g. a table of contents), guided tours and indexed guided tours. Conditional logic structures operate on indices and guided tours to provide cross-entity navigation. Isakowitz et al. (1995) describe the steps followed in RMM as: (1) E-R design, (2) Slice design, (3) Navigational design, (4) Conversion protocol design, (5) User interface design and (6) Run-time behaviour design. RMM ignores analysis altogether, focusing exclusively on design, largely of presentation and navigation. While step 6 does address processing, it doesn’t provide sufficient detail for specifying business rules.

Schwabe and Rossi (1995) developed the Object-Oriented Hypermedia Design Model (OOHDM) which builds a web application in four stages, namely: conceptual design, navigational design, abstract interface design and implementation. Again, there is no analysis stage and navigation design is the focus.

Similarly, the Scenario-based Object-Oriented Hypermedia Design Methodology (SOHDM) of Lee et al. (1999) consists of stages for domain analysis, object modelling, view design and navigation design prior to an implementation stage, ignoring analysis and focusing on navigation design.

Takahashi and Liang (1997) proposed a method for the design of web sites that used both static and dynamic models, ERDs for the former and scenarios for the latter. In this method, E-R modelling is used initially to scope the system under development. Scenario analysis is then used to detect users of the system and how they might interact with the system. The users are called agents and the interactions are known as actions. There is a marked similarity between this stage and use case modelling, where actors are the agents and use cases are the interactions. The next stage in this method is to create a relational data management model for web based information systems (RMDMW) based on the E-R diagram. This diagram is derived from the RMDM of Isakowitz et al. (1995) as used in RMM and involves transforming the entities and relationships into web resources and navigation links respectively. By including an E-R model and use cases, this method is the only one of all the methods reviewed here that support analysis at some level rather than design. However, analysis and design of processing, particularly with the detail of business rules, are still missing from this method.

From the above we can see that extant methods for web-based system design have focussed on design of navigation and have largely ignored analysis, particularly analysis of processing, and have inadequately covered design of processing.

In the next section, we look at the original Business Rules Diagram method (McDermid, 1998), which was used as the basis for the extensions proposed in this paper.

THE BUSINESS RULES DIAGRAM METHOD

There seems to be some disagreement as to what might constitute a business rule (Loosley, 1992). According to Sandy (1996), the range of opinion covers a variety of items such as critical success factors, quality goals and database integrity constraints. Adapting from McDermid (1998), we shall define a business rule as:
A description of the states, conditions and stimuli required to change the state of an information system or its host human activity system so that it will subsequently respond differently to external stimuli, together with the changes required and any associated responses.

McDermid’s original (1998) Business Rules Diagram (BRD) method uses a state-based model of a business rule, that follows a notation similar to flowcharts. The method has two versions of the diagram. The simpler User Business Rule Diagram (UBRD) supports elicitation and may be shown to and verified by, or even developed by end-users. A UBRD contains four explicit constructs, these being states, events, conditions and signals (see figure 1), connected in combinations to specify a business rule. States reflect the status of an object of interest at any given time. 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. 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 that leave the system serve to inform those outside the system boundary about what has occurred inside the system and are called messages.

The full Business Rules Diagram (BRD) extends the UBRD by adding the construct of the Harel blob (Harel 1988), which encapsulates other constructs and is used to model selection or simultaneous action. The BRD is intended for use by systems analysts for abstracting system aspects and reasoning about the requirements.

![Graphic Notation used in the Business Rule Diagram Analysis Method.](image)

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

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

Prior to these studies there were two earlier action research studies, which we will call Study A (Johnstone and McDermid, 1999) and Study B (Johnstone et al., 2000). Study A concerned low-level validation of the analysis method of McDermid (1998) and led to the further development of the BRD analysis method. Study B applied the refined method to a web-based e-journal publishing system. During the study, identification of actors and use cases were incorporated into the method. Needs identified in Study B lead directly to the development of the web-based design notation and their testing in the studies reported in this paper, which we will call Studies C and D. Studies C and D differ from prior studies in that the BRD method is used here to model not just requirements but also design artefacts for web-based systems.

**WEB DESIGN EXTENSIONS TO THE BRD METHOD**

Given that the BRD method captures business logic in a variety of forms and is, in essence, a dynamic, process-oriented, state-based method, it seems appropriate to extend the BRD method to cater for the complexities of web applications. But what extensions should be made and how? This section describes the extensions.

One of the interesting outcomes of Study B, from a modelling point of view, was the separation of navigation logic from processing (business) logic in the diagrams created by the development team. This empirical evidence of a natural dichotomy is in accord with recommendations from other researchers (Bichler and Nusser, n.d.; Conallen, 1999). Separating these concerns while incorporating navigation was a goal of the extensions. A summary of these distinct system viewpoints is shown on figure 2.
The viewpoints are further separated into two distinct diagrams, Process Design Diagram (PDD) and the Navigation Design Diagram (NDD). The notation for the PDD, which is derived from the detailed analysis Process Rules Diagram (PRD), is shown in figure 3. The notation for the NDD is shown in figure 4. However, in practice some mixing of concerns must exist in order to coordinate navigation with processing. This linkage is accommodated in the method by including the navigation states and events shown in figure 2 as constructs in the PDD.

![Diagram of Design Views](image)

**Figure 2: A Conceptual Model of the Design Views of the BRD Method and their Associated Elements.**

The activity of design, especially high-level design, must map the requirements determined in analysis to high-level descriptions of the use of technology. In making this mapping, it is useful if the view and notation used in the analysis artefacts (along with the model and notation used to create them) are carried over into the design with a similar logical perspective. This was a second goal of the extensions. The BRD notation was refined to add navigation-specific symbols, which mirrored the processing-specific symbols already present in the BRD notation. These are shown in figure 3.

![Diagram of Extended Notation](image)

**Figure 3: Extended Notation for the Process Design Diagram.**

When designing the symbols for a design notation one must decide whether the symbols chosen should be technology-specific or not. Given that the technology used in EC systems is not stable because new technologies are being invented at a high rate, we decided not to make the symbols technology-specific. Thus, as shown in figure 3, processing-oriented events are shown with the same rectangle as events in the BRD. An abstract event is one for which the technology is not (yet) chosen. If a technology is chosen, then the technology is given in the name and the rectangle is shown with a shadow. Similarly, the process block (see figure 3) includes the choice of technology, if known (e.g. API, servlet, applet, script etc.), as text within its symbol. Furthermore, the symbols for nodes and links on the NDD (see figure 4) are also not specific to the technology (e.g. HTML vs XML), only to their position (start document vs node document) and their navigation connectivity within the NDD.
McDermid’s (1998) BRD analysis method steps (shown further above) are augmented with the following steps for the production of the new design artefacts. The first two steps concern design of the processing and identifying the needs for navigation. The last two steps design the navigation.

- construct Process Design Diagrams (PDDs);
- refine Event Specification Tables (ESTs);
- construct Navigation Design Diagrams (NDDs); and
- construct Navigation Tables (NTs).

Theoretically, the extended BRD method (process and notation) achieves its design goals of supporting navigation, mapping analysis onto design easily, separating navigation design from process design while still linking them, and supporting specification of technology without using technology-specific notation, as described above. However, any proposed method should be empirically evaluated. In the next section, we describe the theoretical framework for the research.

**RESEARCH METHODOLOGY**

The research framework used to guide this study was that of Nunamaker et al. (1991) as modified by Venable and Travis (1999). See figure 5. System or method development research often follows one of two cycles as shown in figure 5. Whether the area of concern is a type of information system or a system development method, typically theory is first built concerning the requirements, architectures, principles, and detailed structure of the system or method. Theory building had already taken place as described above in the introduction, literature review, review of earlier studies, and rationale for the extensions. Next, the system or method is developed or revised in accordance with the theory. The system building of the web-based system BRD method extensions was described in the previous section. The difference between cycle 1 and cycle 2 in figure 5 is the action taken to evaluate the system or method. In cycle 1, the evaluation is *in situ*, i.e. in real-life contexts, with all the benefits (and research complications) of realism. In cycle 2, the evaluation is in a contrived experiment, which has the advantage of being better able to control for confounding variables and achieve greater research rigour, but the disadvantage of a lack of realism and potentially poorer or weaker relevance.
In order to better assure research relevance, we chose cycle 1. For the research method for the in situ evaluation of the extended BRD method, we chose action research. The web-based system development projects selected were of a large enough size to exercise the method, encounter complexity of requirements, deal with multiple development team members and their clients, and to give some diversity in the participants and the target systems involved.

Checkland maintains that a fundamental axiom of conducting action research is to declare one’s framework (F) in advance of entering the problem situation (Checkland, 1985). Only by explicating this can the research be comprehended within its context and robustness of conclusions justified. The research described in this paper is the continuation of previous research by the authors of this paper and so importantly what is described as the technique here represents current thinking as speculated by the authors about the ‘framework of ideas’ in Checkland’s terms and thus the worthiness of the technique is at the heart of what is being evaluated.

The research questions considered are typical of systems development research. Can the method be successfully applied in the domain for which it is intended? What benefits are realised? What problems are encountered? What workarounds (if any) do the participants come up with when faced with problems? How well do they work?

These steps of the extended method as described above were followed in two action research situation studies (Studies C and D). Two groups of participants were taught the notation and steps of the extended BRD method and then attempted use the method to analyse, design, and build two different web-based systems. They were aided by one of the researchers acting as the group facilitator. The studies and some of the more important results are described in the next section.

THE ACTION RESEARCH STUDIES

In this section, we will describe the environment and scope of the studies as well as the progress made by the teams at each stage of the development. In the first study (C), the client was a regional wine industry association. The participants in the study were a systems analyst with 17 years experience in information systems development (the researcher), five developers or method users (third year university students in a non-computing degree), a main client representative (with minor assistance from two other client reps.) and an external contractor. The problem domain was that of contact management. The intention was to produce a database system to store a complex set of member details with a web-based extension to allow members of the association limited access to their own records.

In the second study - study D (which was conducted in parallel to study C, but over a longer time frame, twelve months versus six months), the client was a university academic with many years of system development experience. The participants in the study were a systems analyst with 17 years experience in information systems development (the researcher), two method users (third year university students in a software engineering degree) and the client. The problem domain was that of digital libraries. The intention of the
method users was to produce a web-based, non-relational (XML) database system to store publications of varied types and to allow public domain access to the system.

Using third year students instead of practitioners raises an interesting question concerning the authenticity of this study vis-à-vis action research. The fundamental question being tested in these studies was to ascertain whether the extended method could be used to model these domains i.e. does it contain the necessary richness of construct to satisfactorily model the problem domains?, are the steps in the method sufficiently clear for the analysis and design to proceed in an orderly fashion? It was considered that the environment of student projects in a university with regular supervisory oversight offered advantage for the monitoring of these questions more than the alternative. As regards the authenticity, both these projects were ‘real’ projects for real users in every sense.

The process used in the studies followed the steps of the method to create a complete BRD as defined by McDermid (1998, described above) and augmented with steps for the production of the design artefacts (also described above).

In both studies, the researcher acted as a facilitator and thus provided guidance only when asked by members of a team, therefore reducing the effect of an imposed world-view from the researcher about what constitutes “good practice” or “the solution to the problem”. In order to track the progress of the studies, each team kept minutes of project meetings, dated the production of BRD artefacts and the researcher kept separate field notes containing reflections about the products and social processes that were the outputs of the studies. Members of the teams in both studies were interviewed separately by the researcher at two key intervention points, the completion of the analysis phase and the completion of the design phase of each system. These interviews were semi-structured and taped for later analysis by the researcher.

Given that the focus of this paper is on design, a brief summary will be given here concerning the analysis models produced by both teams. The team in Study C articulated 107 business rules across 12 major functions covering many aspects of the operation of a contact management system. A use case diagram was generated by the team during this stage and was used to classify the business rules and served as a top-level context diagram throughout the remainder of the study. Use case diagrams were introduced into the BRD method as a structuring mechanism in a prior study (Study B). The participants were able to produce a coherent requirements specification that was signed-off (and presumably understood) by the client.

Study D was more complex with the team articulating 168 business rules across 18 major functions covering many aspects of the operation of a digital library system. As in Study C, a use case diagram was generated by the team during this stage. The method users in Study D were also able to produce a usable requirements specification that was used as the major input into the design phase of the system being developed.

LESSONS LEARNED FROM THE STUDY

Modelling Concepts Unique to a Web based System

Both teams were able to separate navigation from process, with the team in study D being somewhat more successful – perhaps due to their background (software engineering as compared to multimedia). Both teams also used most, but not all, of the design constructs available to them in the BRD method. Neither team used the NDD composition link to model frames, which was surprising, especially given that the participants in study C were multimedia students who were initially more concerned with layout rather than process.

Client Take up of the Method

This research programme has conducted prior studies with method users who were either experienced business analysts (Study A) or web developers (Study B) or academics with doctorates (Study B). A valid criticism is that those studies proved only that the BRD could be taught to experienced, motivated professionals but there is no evidence to suggest that more inexperienced users could successfully articulate and build models which can then be transformed into working implementations. The results from these two studies suggest that even users with little or no real-world experience can be taught the rudiments of the BRD method and can use the method to articulate models of real-world systems that are sufficiently realistic to be a) acceptable to a client and b) implementable.

Transition issues from Analysis (the PRD) to Design (the PDD)

The method users in Study C were able to produce reasonable analysis models (a use case diagram, business rules, ESTs and PRDs) that were accepted by the client as being representative of the planned business processes of the firm. Interview records indicate that the team was able to use the models to reason successfully
about expected system behaviour. In contrast, the design models produced by the same team were not of the same standard of usefulness. The most likely hypothesis is that the team didn’t really understand the difference between analysis and design, which is reasonable given the supporting evidence found in the interview records when team members were asked firstly, how they created design artefacts from the analysis diagrams and secondly, how they used the design to produce an implementation.

Another issue that arose was that in Study C, the method users were unable to complete the project and therefore did not produce an implementation for their client. This was partially due to internal project team problems (which might be expected of undergraduates) and partly due to a lack of requisite implementation skills within the team. A particularly interesting outcome from this study from a research viewpoint is that the research cycle and the project cycle were not precisely aligned. The outcome was that just because the team could not produce a deliverable that satisfied the research goals, that did not mean that there was no ethical responsibility on the part of the researcher to provide a system for the client.

In contrast, the method users in Study D had much less trouble in moving from analysis to design. This could be because they had a clear intention of an XML-based end-product, perhaps their client had a better idea of what he wanted or possibly they were more sensitised to the difference due to their background (software engineering students). Interview transcripts show that the method users understood the purpose of design (as distinct from analysis) and were comfortable using the extended design notation of the BRD method.

Figure 6 shows how the team in study C melded navigation and process by wrapping navigation (e.g. the S1-S2 and S4-S5 transitions) around key business processes (the centre blob). The salient feature of the diagram is the linkage of navigation elements in a process context (in figure 6) to navigation elements in a navigation context (figure 7). When examining both diagrams, the links can clearly be seen as artefacts that have been converted between representations e.g. in figure 7, the documents “login page”, “forgot password”, “privacy statement” and “update/edit details” have been derived, either directly or indirectly, from the navigation constructs present on figure 6 (i.e. states S1, S2, S4 and S5).

It is should be noted, however, that the models produced by the method users in Study C were not perfect. An example of an error in figure 6 is the state S3, which is a navigation state, should be outside the process blob. Similarly, event E4 should have been placed inside the process blob. Given that a correct navigation design diagram could still be produced from an erroneous source suggests that the BRD method is resilient to errors under some circumstances, although it is acknowledged that there is no direct proof of this effect and there could have been other factors that contributed to this effect.
Figure 6: Process Design Diagram for the “Authenticate” Use Case (Study C).

Similar models were produced by the method users in Study D, although the models were, in this case, slightly more complex and had fewer errors (i.e. were more correct with respect to the syntax and semantics of the BRD method). The notion of embedding process within navigation was also evident in Study D.

In web-based systems development there is often a need to model that many node documents that appear in the navigation view need to link back to a root document. This is handled in the BRD method by specially decorating a node document, which has the advantage of reducing the number of arrows drawn on a navigation design diagram and thus increasing clarity. The method users in Study C took this idea a step further by introducing the concept of a “local group link”. This can be seen as the red (light grey) squares denoted in the documents “renew member”, “update/edit details”, “forgot password” and “new members” which indicates that all of these documents contain a link to the “privacy statement” document, denoted by red (light grey) text. Whilst this demonstrates some capacity to innovate, there are other simpler ways to achieve the same result without resorting to special colour schemes e.g. grouping via Harel blobs (which already exists within the BRD method).

Space does not permit an example or a detailed discussion of the navigation table, which is an alternate (but richer) model of the navigation design diagram, but briefly, method users in both studies were able to produce and use the appropriate navigation tables. Again the navigation table in Study D was significantly more complex (a 27x27 matrix) than that produced in Study C (a 9x9 matrix).

In Study C, the mixing of process and navigation constructs in the PDDs could have been an attempt by the team to meld the two viewpoints together, which would have been a laudable aim. It is more likely though, that this blending was evidence of the team's inability to see the relationship between analysis and design using the BRD method. There is some evidence for this claim to be found in the interview records of Users H and J respectively.

(Researcher) Were there any problem areas with the notation?

(User H, Project Manager) Well, I guess we felt that the PDDs was [sic] repeating the software requirements specification document a little bit. Seeing as a lot of our work was...had nothing to do with navigation, about half of it would have had nothing to do with navigation in a web sense. So I guess there was a bit of a feeling that we were repeating something we’d already done and it seemed a little redundant at times.
(User J, Systems Analyst) Actually, I found the one for the requirements specification was pretty easy, except I found the PDDs a little more confusing as a lot of it seemed to be doubling up on what the BRDs showed.

Figure 7: Membership Area Navigation Design Diagram from Action Research Study C.

Another aspect of this interaction is to consider the participants as interpreters where the interaction produces data and/or theory. In this context, the team took some theory about how to construct a design from the researcher (the BRD method) and a set of data (the artefacts generated from the client meetings etc.) and produced an interpretation which did not quite fit the theory, thus they modified the method to fit their Weltanschauungen.

It is interesting that both teams chose to "wrap" navigation around key process elements in the design. Neither team was instructed to do this by the researcher, it is possible that both teams deduced that this was a correct approach from examining figure 2. A comment from User N (Study D) suggests that this was not so, however:

(User N, Analyst/Programmer) I'd like to say we changed the navigation, put some up [above] and some down [below] between the process.

(Researcher) Ok, so we're looking at figure 2 of your design, the search PDD?

(User N, Analyst/Programmer) Yeah. At first when we drew the PDD, right, we put the navigation, when we started doing the process design diagram always putting navigation below the process, then we realised it would be better to analyse the situation how it would be and maybe some navigation would come before the process and some after the process. For instance, with "search", if you go to the navigation page to search then a process takes place and then you get a result, so we separated that a bit.

(Researcher) Why is this better than having the navigation just at the bottom of the process blob?

(User N, Analyst/Programmer) 'Cause, like, if someone looked at the system, I mean would look at the design diagram, it was starting with a process and then coming down to the navigation saying "go to page" or "search page" or "start searching" and the results page and the process was above it. I mean, when you start searching...after that is when the process takes place and that's when you get the result, after the process has taken place so I mean we found it better to put it that way.

Conclusions about the technique

It is important to remember that the studies reported above are only part of an ongoing program of research into this technique. In other words the results discussed here, on their own, only serve to re-affirm or disconfirm to some extent, previous conclusions and speculations. Clearly the lack of overall success in the completion of project C might point to weaknesses in the technique/method, weaknesses in the individual members of the team, weaknesses in the project environment and infrastructure, politics or any combination of these. That is why due and careful examination of the problems that arose have to be reflected upon by the researchers and
their likely cause or causes speculated and further explored. In the case of project C, the supervisor was able to
establish that there was no technical inadequacy of the technique itself in terms of its ability to express a correct
solution to the domain investigated. Rather the problems were related more to the inability of the team members
(for whatever political, training, background or cognitive reasons) to create a correct model. Project D on the
hand was considered by the client and researchers an overall success and confirmed much of the speculations
about the technique. In other words, while there are many social aspects of these studies to reflect on, the ability
of the BRD to model requirements and specify web-based design from a technical perspective, was re-affirmed.

SUMMARY

We suggest that further detailed research is needed on the BRD method and other diagrammatic modelling
tехiques and methods in order to further explore their use in practice. Across-method research is also needed
before the findings of this research can be generalised to other methods. Future work will include a comparison
of the BRD method with the UML as well as other studies designed to more fully explore the design-
implementation interface. Having said that, the design issues highlighted in this paper have clearly
demonstrated a compelling argument for research into articulating web-based analysis and design. Given the
paucity of publication in this area, this paper represents a positive action-orientated response to addressing that
gap.

REFERENCES

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


63-70.

Group.

Journal of Electronic Commerce Research, 1(1).


Design, Communications of the ACM, 38(8), 34-44.

in Hope, B. and Yoong, P. (eds.) Proceedings of the 10th Australasian Conference on Information

Electronic Commerce with Business Rules, in Gable, G. and Vitale, M. (eds), Proceedings of the 11th

Information & Management, 36, 121-38.


Technology, Perth, Western Australia.


Conference on Information Systems, Hobart, Tasmania, Australia.


**COPYRIGHT**

© 2003. The authors assign to ACIS and educational and non-profit institutions a non-exclusive licence to use this document for personal use and in courses of instruction provided that the article is used in full and this copyright statement is reproduced. The authors also grant a non-exclusive license to ACIS to publish this document in full in the Conference Papers and Proceedings. These documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. Any other usage is prohibited without the express permission of the authors.