



VICTORIA UNIVERSITY
MELBOURNE AUSTRALIA

A development-oriented DSS evaluation approach: A case demonstration for conceptual assessment

This is the Published version of the following publication

Miah, Shah Jahan, Debuse, Justin and Kerr, Don (2012) A development-oriented DSS evaluation approach: A case demonstration for conceptual assessment. *Australasian Journal of Information Systems*, 17 (2). pp. 43-55.
ISSN 1039-7841

The publisher's official version can be found at
<http://dl.acs.org.au/index.php/ajis/article/view/694/535>
Note that access to this version may require subscription.

Downloaded from VU Research Repository <https://vuir.vu.edu.au/21591/>

A DEVELOPMENT-ORIENTED DSS EVALUATION APPROACH: A CASE DEMONSTRATION FOR CONCEPTUAL ASSESSMENT**Shah Jahan Miah**School of Management and Information Systems
Victoria University
shah.miah@vu.edu.au**Justin Debuse**Faculty of Arts and Business
University of the Sunshine Coast
j.debuse@usc.edu.au**Don Kerr**Faculty of Arts and Business
University of the Sunshine Coast
d.kerr@usc.edu.au**ABSTRACT**

As a subset of the Information systems (IS) discipline, Decision Support Systems (DSS) development research needs a broader conceptual lens in order to evaluate design qualities specific to DSS. This paper presents a development-oriented approach for evaluating DSS applications. Based on a design science mental model proposed by Peffers, Tuunanen, Rothenberger and Chatterjeea (2008), our study outlines a new application area of this conceptual approach through a case demonstration of DSS development through various design checkpoints. To assess the proposed development-oriented evaluation strategy the paper also describes a qualitative investigation that has been undertaken to assess the conceptual approach with target industry DSS users and developers. The findings suggest that this strategy is of use for improving the overall design qualities in every phase of DSS development, and could be useful for evaluating DSS design with target users and DSS designers within a socio-technical design context.

Keywords: Design Science Research, Evaluation Methods, Decision Support Systems

INTRODUCTION

There is a long tradition by Decision Support Systems (DSS) researchers of employing various information systems (IS) methods to evaluate DSS development. Evidence of this can be viewed through many DSS development studies (Muntermann, 2009; Purao and Storey, 2008). The evaluation of various specialised DSS applications such as ES (expert systems) and KBDSS (knowledge-based DSS) have previously employed a validation and verification based strategy. However, it is suggested by Phillips-Wren, Mora, Forgionne and Gupta (2009, p. 643) that classical IS evaluation frameworks and models “*lack a holistic worldview that considers jointly the organisational, designer and builder criteria of interest*”. Using such criteria can be problematic for evaluating DSS, as these can “*proactively impact the process of decision making as well as the outcome, by providing for example, real time response, distributed architectures and autonomous*

behaviour to support the decision makers". Arnott and Pervan (2005) supported this argument by differentiating the key attributes of the DSS from traditional IS design, as it "represents a different philosophy of support, system scale, level of investment, and potential organisational impact. They can use quite different technologies and may support different managerial constituencies" (p. 68). As the development of such computer-based personalised support system is crucial in a specific problem domain, our study addresses particular boundaries of IS application design and development through the differentiation of DSS from IS.

Good DSS design requires having a clear concept of the nature of the target decision problem and a well-defined strategy of how to adequately support the decision process (Arnott, 2006). Classical DSS development methods such as evolutionary prototyping provide little support to system developers on how to proceed, as they do not have checkpoints that can be used to evaluate design qualities throughout development. Arnott (2006) also described another problem for a DSS developer, namely how to conceptualise aspects of the decision task that need improvement during iterations of the evolutionary development process. We contend in this paper that a DSS developer needs to employ approaches that are different from traditional IS evaluation methods because DSS development includes factors that are significantly different from the traditional IS approach. In other words, the methods used for DSS evaluation do not have the same goals as those in IS evaluation. We aim to address this problem through a design science approach based on research conducted by Peffers, Tuunanen, Rothenberger and Chatterjeea (2008). These authors proposed a conceptual model in the design science paradigm that has six phases, namely: define problem, define objectives, design and development, demonstration, evaluation and communication.

Arnott and Pervan (2008) have suggested that DSS research has a long history of using design science research, and other studies also indicate that the rigour of DSS research needs to be improved (Arnott and Pervan, 2005, 2008). In addition, Mackrell, Kerr and von Hellens (2009) proposed a socio-technical design approach for enhancing the human perspective and situational practicalities of decision making in a specific problem domain. Motivated by these suggestions from Arnott and Pervan and the socio-technical design for IS development, we attempt to address the complex requirements of DSS evaluation through the mental model approach proposed by Peffers et al. (2008), using the six-phase checkpoints throughout DSS development. Therefore, this paper describes a qualitative research study designed to investigate the applicability of Peffers et al.'s model for evaluating decision support systems within a socio-technical context. In Mumford's (2000) definition, "*socio-technical design is an approach that aims to give equal weight to social and technical issues when new work systems are being designed*" (p. 125). Within this context, this paper discusses how activities in different development phases or checkpoints can be used to assess successful design.

This study assesses this comprehensive evaluation approach¹ within a decision support systems (DSS) problem context. Our findings suggest that the evaluation approach can be viewed not only as a basis for further knowledgeable actions in organisations, but also as a new lens with which to evaluate DSS in a broader, more practical perspective. We focus on applying the design science mental model to evaluate various development stages of DSS. Our key argument is to illustrate how a design science-based approach offers a better strategy for DSS evaluation, since it can take the entire development activities into consideration. Within our study, we verify the appropriateness of the method-based

¹ The approach was presented initially at the 5th Conference on Qualitative Research in IT (QualIT), 29-30 Nov 2010, Brisbane, Australia.

mental model in a real-world DSS development environment. Utilising an interpretivistic approach to capture independent opinions of the target stakeholders group, we use a forestry pest management DSS as a case demonstration. This type of evaluation is useful since the DSS has the potential to provide rapid, simple answers to important problems within an industry.

The socio-technical design science approach presented by Gregor and Jones (2007) defines the structural components of IS design research as: “(1) purpose and scope, (2) constructs, (3) principles of form and function, (4) artefact mutability, (5) testable propositions, (6) justificatory knowledge (kernel theories), (7) principles of implementation, and (8) an expository instantiation” (p. 312). In a similar conceptual discussion, McKay, Marshall and Heath (2008) indicate multiple views of artefact design using aspects of the research outlined by Iivari (2007) and Carlsson (2006). Both approaches imply that IS design within the socio-technical context requires a broad approach as there are various significant components. A key component of an evaluation strategy should be the examination of the developed design within the context of the problem domain and, since most DSS development problems result from poor identification of end users' needs (Arnott and Pervan, 2008), end users should be able to determine whether the solution is “good or bad”. A conceptual approach such as a mental model can also give a template with checkpoints for end user and designer to help them evaluate the design under a broader context. Therefore, a key question for this study is: *Does the design science-based evaluation approach address DSS development qualities in a socio-technical context?*

The rest of the paper is organised as follows: the next section provides details on the background literature of the design science method and IS evaluation issues. Subsequently, the problem context and method used for conducting the study are described. Next, the conceptual approach and its application for evaluating artefacts are defined. The following section provides details of the case evaluation and quantitative findings from our empirical investigation. Finally, the conclusion summarises the key contributions and further research from this study.

THEORETICAL FOUNDATION

Design science methods

The primary focus of conventional methods in design science research covers two areas: one based upon innovative design theory for a technical artefact, focussed upon the quality of the artefact (Hevner, March, Park and Ram, 2004; March and Smith, 1995), and the other based upon a design within technical and social phenomena, focussed on design realities such as the impact of the artefact on an organisation or society (Gregor and Jones, 2007; Iivari, 2007; McKay and Marshall, 2007). Both views may have implications for guiding design practices, but in DSS design most of the solution design needs to address problems in decision making that are unstructured or semi-structured. The design innovation in DSS covers four areas: designing methods and instruments; designing interactive computer-based systems for improved decision support; designing decision-maker oriented process; and designing flexibility in separating data and models in DSS applications (Carlsson and Turban, 2002). In addition, in most of the previous DSS development approaches, researchers, developers and users have suffered from a lack of practical knowledge of how the intended system design fits the problem context, such as rural industries (Miah Kerr and Gammack, 2009; Miah, 2008; Kerr and Winklhofer, 2006; McCown, 2002; Cox, 1996).

The design science approach within the DSS field is not a new research area, and many IS researchers have described the use of design science in DSS development. Examples include Muntermann (2009) and Arnott and Pervan (2008), who analyse DSS development in the context of a design science methodology developed by Hevner et al. (2004). They found that, although design science has been

widely used in DSS research since the 1990s, a key shortcoming was the paucity of evaluation methodologies and supporting theory. This study addresses the identified gap by applying an evaluation methodology, based on design science theory, to the phases of development involved in producing a DSS. Focus groups, the primary evaluation approach suggested by Arnott and Pervan (2008), are used in this study. This broader use of the design science method can be considered a useful lens for evaluating DSS applications as it identifies the effectiveness of each development phase through various checkpoints.

Issues in evaluation approaches

Many approaches have been used in the evaluation of IS, and considerable overlap exists between them. Each approach involves different features for measuring IS qualities. For example, Hirschheim and Smithson (1988) (described by Jackson and Sulaksono, 1998) include the system benefits, which are examined by Hitt and Brynjolfsson (1996) in the areas of customer value, profitability and productivity; similarly, the technical performance of the system is measured, as performed by DeLone and McLean (2003). Hitt and Brynjolfsson's approach does not cover perspectives such as information quality and how well the system is provided as a service, unlike DeLone and McLean's (2003) model. Further, existing approaches do not appear to include evaluation of the development processes used to produce the system, although Jackson and Sulaksono (1998) note that Hirschheim and Smithson (1988) include the maintainability of software to be part of its technical performance.

When considering DSS evaluation, Forgionne (1999) suggested a model that could measure effectiveness based on the decision making process. Parikh, Fazlollahi and Verma (2001) proposed a similar view that used four evaluation criteria based on decision quality, decision making satisfaction, decision makers' learning and decision makers' efficiency. However, Phillips-Wren, Hahn and Forgionne (2004) commented that these findings should include information about the outcomes in terms of decision process improvement and any gains in the users' ability to handle decision problems. This implies that the DSS evaluation strategy should focus on the process, users and their effectiveness in handling decision-making scenarios. This finding also applies to specialised DSS development such as ES and KBDSS. As described earlier, conventional evaluation approaches for specialised DSS development are based on the *V & V* (verification and validation) approach (Mahaman, Passam, Sideridis and Yialouris 2003; Mansingh, Reichgelt and Bryson, 2007; Juristo and Morant, 1998; El-korany, Rafa, Baraka and Eid, 2000). Verification is described as "building the system right", that is, a system appropriately implements its specification. Validation is described as "building the right system" (Mahaman et al., 2003). For evaluating ES, Mansingh et al. (2007) conducted the validation approach by comparing the system outputs with a case solution by human experts. The evaluation was based on examining the effectiveness of the user interface and the system's recommendations. This type of evaluation is more appropriate for domain-specific system development where the system serves specific user groups. Such strategies are not helpful for achieving broader objectives for DSS developers, such as determining the quality of the decision or evaluating the design from the users' point of view.

PROBLEM CONTEXT AND METHODS

Study context

To determine the applicability of our framework, we evaluate the conceptual approach in a real-world DSS development environment within a rural industry. The problem context we used in the study is decision making in forestry pest management. This is a very important problem area, since pests have the potential to render plantations unviable (Carnegie, Stone, Lawson and Matsuki, 2005), and the choice of treatment regime is crucial in minimising this. The DSS examined within this study allows

users to input a range of characteristics of a forestry site, such as the spacing of the trees, pest species identified and proportion of foliage damaged. The system then outputs a recommended action to address the identified problem, and where possible allows users to access pdf files containing further information such as photographs of symptoms and description of insecticide treatment. To capture this situation, we developed a prototype following the six activities of the proposed approach. This allowed us to evaluate the usefulness of the proposed evaluation framework from the perspective of forestry scientists; a key factor in the evaluation of usefulness was the familiarity of the scientists with the system and its application domain.

Methodology

The aim of our interpretive approach is to capture a range of independent opinions from a target group of stakeholders. We focused on the key components of DSS evaluation from a socio-technical design perspective, such as stakeholder orientation, process changes and key evaluation criteria according to the situational needs. We therefore evaluated our proposed approach using interviews. We used four individuals for the interviews and asked participants to determine how useful they perceived the evaluation checkpoints to be; these checkpoints, together with their stages, are:

1. Outline the decision problem
 - a) Problem importance
 - b) Problem suitability for decision makers
 - c) Problem complexity / simplicity
2. Define design objectives
 - a) Whether quantitative or qualitative measures are to be used
 - b) Appropriateness of objectives
 - c) Resources required by design objectives
3. Artefact design and development
 - a) Design and development approach used
 - b) Measures used to determine the innovative features of the system
4. Identify design context
 - a) Determine the context in which the system is to be used and further tested
5. Measure effectiveness and efficiency
 - a) Effectiveness and efficiency of system within its application context
 - b) Whether effectiveness and/or efficiency meet target requirements
6. Communication of results
 - a) Determine how the outcomes are to be presented
 - b) Determine whether the communication structure is appropriate for the target audience
 - c) Determine whether the system outputs match discipline knowledge.

PROPOSED EVALUATION APPROACH

We used the framework shown in Figure 2 (adapted from Peffers et al., 2008) to evaluate DSS development because it is able to incorporate product, practice and process perspectives within a socio-technical context.

The evaluation begins with problem identification; complexity is also determined to establish whether development of a DSS would be infeasible or overly trivial. The extent to which the developed solution addresses the identified problem is then determined, using qualitative or quantitative measures. The approaches used to build the system, together with the quality of the knowledge embedded within it, are then evaluated. Next, an assessment of how well the solution works within its problem context is made. Following this, the effectiveness of the system, including for example the patchiness of data availability and its efficiency (such as how swiftly it allows decisions to be made),

are evaluated. We provide an option to return to the third step if performance is unsatisfactory; otherwise, an assessment is made of how well the system communicates its results, covering areas such as whether they are related to domain knowledge or their form is appropriate for their target users.

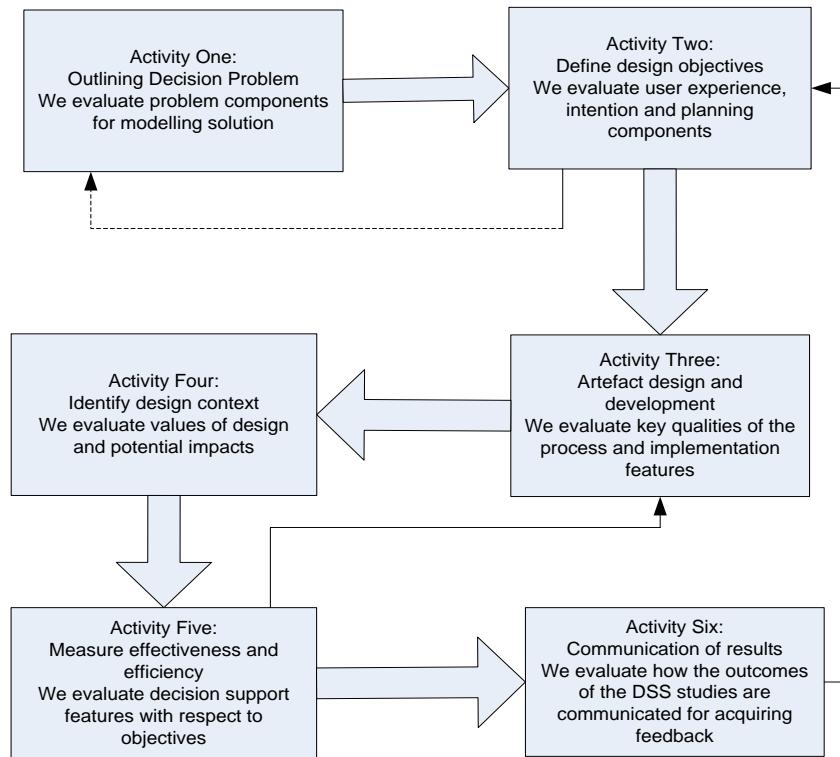


Figure 1: Conceptual evaluation approach based on the framework proposed by Peffers et al. (2008)
(extracted from Miah et al. 2009)

CASE EVALUATION AND FINDINGS

In the DSS design research domain, Arnott and Pervan (2008) suggest that a design artefact can be a construct, model, method or instantiation. March and Smith (1995) define a method to be “*a set of steps (an algorithm or guideline) used to perform a task*” (p. 257). This implies that the proposed evaluation approach may be considered a design artefact. Hevner et al. (2004) proposed five distinct methods for evaluating designs and these are shown in Table 1.

We considered the descriptive method to be the most appropriate for our problem context, and so this was used for our study. Our aim was to capture the utility of the design by constructing a detailed scenario. We focused on the application of the proposed evaluation approach in the practical context to build a convincing argument to show the utility of the proposed approach. The utility of the proposed application of the approach in relation to the existing knowledge base of scientists using the system was measured, along with its usefulness within the target problem context.

Method	Description
Observational	Where designs are studied within projects
Analytical	Where qualities of the design such as its performance and complexity are measured
Experimental	Where the design is evaluated in a controlled context
Testing	Where designs are checked for errors
Descriptive, comprising “Informed Argument”	Where the utility of the design is measured using existing knowledge, and “Scenarios”. The utility of the artefact is described within a target problem scenario

Table 1: The five methods for design evaluation, adapted from Hevner et al. (2004)

Members considered scientists end users	Members considered plantation managers end users
Stage 1: Problem definitions	
<p><i>– The importance of the problem is separate from system quality, and complexity would not detract from the need for system development; it would just make it harder to do.</i></p> <p>One group member commented: “I don’t think you’d do a bad job of creating a product or a less important system just because it’s less important would you?” However, respondents considered “need for system development” in place of “system quality”</p>	<p><i>– If the system is to be developed it must be important and it must be of high quality, The problem must be sufficiently complex to warrant development.</i></p> <p>This is considered a circular question, as they are in favour of system development.</p> <p>Simpler problem solving would depend on the end user (more about context): “We’re deliberately tailoring the questions so that they can answer them”.</p> <p>The system is deliberately tailored to make it simpler, and remove the onus on users to determine what causes damage; instead, they can focus on the extent and type of damage. The paper version would be an alternative if simpler. Problem complexity is important: “because you don’t want to turn the end user off because it’s ... too complex for them”.</p>
Stage 2: Define design objectives	
<p><i>– Any assessment of system quality would be very useful. Resources used is not a good measure of usefulness.</i></p> <p>The following comment was received: “It would be useful to be able to measure whether your system is working, qualitative or quantitatively, if that’s what the question’s asking, whether you want to be able to evaluate it once it’s produced”. This needs addressing before development.</p> <p>Objectives need to be well defined and realistic; it is very useful to have realistic objectives: “I think it’s about having ... defined objectives appropriately, but ... this is one of those questions that comes in before you would develop it, isn’t it again? But certainly appropriate to have objectives</p>	<p><i>– The more resources spent on the system, the higher the quality. Quantitative assessments are very important.</i></p> <p>It is important to know which assessment type is used, as quantitative assessments are required and are very important: “I think this is very important because the end users want to know ... with some degree of confidence that it’s recommending the right action”.</p> <p>This appears to be a circular argument – they would not be discussing it if the system were not wanted, unless it was a random problem: “we’re not ... approaching this as a random problem”.</p> <p>Usually the more resources spent on a system the higher its quality: “Usually if you put more effort into</p>

<p><i>and to define them, and for them to be realistic</i>". As long as the system is useful at the end of it all, resources used is not a good measure. Easier systems will take less time to develop: "<i>If it's just an easier system it takes less time to develop</i>".</p>	<p><i>things they're usually a lot better, though it depends on the complexity of the problem as well.</i>"</p>
Stage 3: Artefact design and development	
<p>– Leave the design and development to the experts. Would not know, and would leave this to the experts. Does not matter if outcome is the same. "<i>I'm not sure I'd know enough about the options ... As long as the outcome was the same</i>". Prefer a working system to a flashy one: "<i>I'd rather have a system that would work than a system that was ... flashy and uploadable ... as long as it worked in one format</i>". If the system worked then innovative features would be excellent: "<i>If it could be transferable to ... the PDA system for example then ... that would make it ... a little bit more flexible</i>"; "<i>that's something that's going to have more impact on the grower themselves isn't it than us</i>".</p>	<p>– Leave the design and development to the experts. Prototyping sounds better if done by someone who can do it properly, but the approach is unimportant other than for implications such as budget or time constraints: "<i>I think the most important thing is that it's honed to the problem that we want and that it's done by somebody who can develop it properly</i>". Innovativeness is important – in forestry it is critical to use some hand-held device out in the field – if it became complex rather than simple it could be a determinant for use on a hand-held device: "<i>from a forestry perspective ... an innovation such as being mobile would be really important. Anything to make the field guys' work easier</i>".</p>
Stage 4: Identify design context	
<p>– Context is very important. Important – specific to plantation management. If it is already developed you could have more flexibility: "<i>In terms of developing it I think the context's important, but if you have something that's already been developed then you could use it in a range of contexts, so it would have flexibility</i>".</p>	<p>– Context is very important. Context is very important – the system is of no use in other agricultural contexts: "<i>It's purely specific to this type of forestry</i>".</p>
Stage 5: Measure effectiveness and efficiency	
<p>– Both effectiveness and efficiency are very important. Both effectiveness and efficiency are very important: "<i>Do effectiveness and efficiency equal accuracy? [Interviewer responds that that would be part of it] ... very important</i>".</p>	<p>– Scientists want the system to be as accurate as possible, but effectiveness is equally as important. Efficiency was not considered as important unless it was problematic, such as the system crashing all the time. Buggy or inaccurate systems would not be used in the field: "<i>guys in the field, if it's buggy and it's always collapsing they won't use it, they'll just go 'oh, I can't be bothered with that' ... if it's not giving accurate results they'll do the same thing</i>". Efficiency was not considered as important as effectiveness; if the system required a lot of power and kept crashing this would be problematic: "<i>if they were using it on their mobile phone and it required a lot of grunt for example, and it kept crashing then ... they wouldn't be bothered with that</i>". Scientists want the accuracy as high as possible as there is so much error in the collected input data: "<i>from a scientist point of view I think you want as high as possible, because you know there's already so much error in the way you collected the data that goes into it</i>". The manager thinks differently, but this</p>

	is being answered from the scientists' viewpoint. The system should work adequately and not crash: " <i>it's not so important to me either whether there were targets related to that [efficiency], I think it's just as long as it works adequately and it's not crashing</i> ".
Stage 6: Communication of results	
<p>– Communication of results was considered very important.</p> <p>Very important: "<i>I was looking at ... how it looked on the screen ... rather than the accuracy of the information it gives you</i>".</p> <p>"<i>that's what it's all about</i>". "<i>To me, the jargon can be communication structure</i>".</p> <p>A system that provides bad results would not be useful: "<i>I don't think there's much use in having a product if it creates advice ... that isn't relevant</i>".</p>	<p>– Communication of results was considered very important – it has to be accurate. The end user just wants the answer.</p> <p>The less output produced the lower the system quality. This is a measure of accuracy – need directed information: "<i>It's a kind of measure of whether those outcomes are likely to be accurate ... If it's only the pdf, then they've got to read through all that and find out whether there are any management outcomes, whereas if the thing states outright you should do this and here's a bit of extra information</i>".</p> <p>Bad structure reduces system quality; for example, salinity maps have been rejected by end users: "<i>I've seen whole projects looking at salinity where they produce these beautiful maps, and then the landholders go 'no, not using that'</i>". Users just want the system to tell them the answer: "<i>Just tell me the answer</i>".</p> <p>Very important: "[Interviewer prompts: if the system says you do action A, and the literature says no, do action B] <i>That's ... not useful</i>".</p>

Table 2: Participants' comments on the proposed mental model for DSS evaluation

Interview recordings were transcribed and the open coding method employed to break down and categorise the data; subsequently the axial coding method was employed to make the necessary connections to the different categories and to integrate the themes. The transcripts were relatively small and the themes were easy to identify; thus, comprehensive analysis using software packages was not necessary.

The procedure for evaluating the approach consisted of three steps. Firstly, we discussed the proposed evaluation approach through a prototype DSS development in the target problem area (forest pest management). The initial discussion helped us to introduce key benefits of the checkpoints and discuss their relevance to the target DSS design. This meeting consisted of three forestry scientists who develop DSS applications for their use. Secondly, this prototype was demonstrated. Thirdly, by means of a questionnaire, we gathered opinions from scientists and end users on different evaluation checkpoints based on the activity. Finally, we conducted a phone interview to acquire inside details of their comments, and recorded the conversation.

Four scientists were interviewed in two groups of two, focussing upon the application of each checkpoint to the prototype DSS. The first interview was done through speakerphone and the second face to face. We illustrate the captured opinions in Table 2 above.

DISCUSSION

The key argument of this study was to evaluate how a design science-based approach may offer a superior strategy for DSS evaluation, since it takes the entire development activities into consideration. The study evaluated a development-oriented evaluation approach for DSS design to

address the call by Arnott and Pervan (2008) to improve DSS design research; Peffers et al.'s approach was used as a template to demonstrate its applicability for DSS evaluation. We contend that this study justifies the application of the approach in the defined theoretical and practical context of DSS evaluation.

We argue that the approach can evaluate whether the entire development and design cycle meets the requirements for DSS development, since it allows us to evaluate properties through various checkpoints from all phases and ensures that the design meets requirements within the context as outlined by stakeholders. This is particularly useful since DSS development is highly iterative, using techniques such as evolutionary prototyping. Because of the need for many iterations and the high level of end-user involvement related to individual context, we consider DSS evaluation to be sufficiently different to standard IS evaluation approaches to warrant this alternative method.

Our approach also covers many of the concerns about DSS development with respect to the social aspects, such as a lack of end-user uptake, rather than concentrating on verification and validation evaluation processes that concentrate on developers' design and technical interventions (Mackrell et al., 2009). The contribution of this study is to enhance the theoretical foundation of DSS evaluation using design science knowledge as initially proposed by Peffers et al. (2008). The proposed approach captures both the development processes and context to enable the evaluation of the artefact within an organisation. This approach therefore identifies a range of issues with development, from the problem identification through to communication of the solution. As discussed previously, we used a socio-technical lens to evaluate a design, in which perspectives such as users, process, practice and product were covered. It is hoped that this approach will also improve the adoption of DSS by end users, because we consider that it will help provide practical knowledge of how the intended system design fits the problem context for both the developer and end users. We believe this will address many of the concerns about how the intended systems fit the problem context as outlined by Miah et al. (2009), McCown (2002), Cox (1996) and Kerr and Winklhofer (2006). This is because the problem identification stages are included in the evaluation, as opposed to previous DSS implementations, where the problem definition was considered a given by the developers but not necessarily by end users.

Practically, the findings suggest that all of the six phases were important. Respondents considered some aspects of the early phases, such as the problem importance and the appropriateness of its objectives, to be self-evident; the relevance of system complexity depended on its target users. However, what is self evident to some is not necessarily so for others, and previous DSS solutions have suffered from a lack of context to the problem at a managerial level Kerr (2004). Therefore we consider an evaluation of these early phases critical, especially at the level of the end user.

The respondents' views on the types of quality measure varied, though both groups agreed that this area was important. Similarly, whilst one group viewed the end product quality as more important than the resources it consumed, the second argued that the two were somewhat interrelated. Obviously the expenditure of resources is very important and is critical for the viability of the DSS project; however, we must infer from the response from the first group that they would consider a high quality DSS with a more limited scope as a more useful product than one using more limited resources because the scope of the project was too big.

The design and development approach, though important, was seen as a matter for system development experts, and this is appropriate. However, it is also very important that design and development criteria are fully understood by both parties, and that communication channels are effective. The innovation measures were considered important, particularly if they represented vital features, such as portability within the problem domain, via mobile platforms in the field.

The later phases were considered to be particularly important in the problem context. All respondents considered the design context to be crucial. In a similar vein, effectiveness and efficiency were considered very important, with respondents suggesting that a system that was not effective, or with efficiency issues impacting upon effectiveness, could not be used. Unsurprisingly, communicating results proved to be the most important phase, in which information quantity, quality and simplicity were considered the key to solution design completion. All respondents considered it very important to ensure that the results were accurate and reflected current scientific thinking. This is an important issue that has been problematic in the past (Cox, 1996), as end users often want a simplified version of the answer; the goal is to ensure that the results shown are simple enough for managerial end users to understand without being a "Black Box" solution as described by Cox (1996). This means that the logic involved in obtaining the solution needs to be transparent to the end user. It is hoped that the approach described here will allow this to happen through the evaluation of all stages of development.

CONCLUSIONS AND FUTURE RESEARCH

This research proposed a development-oriented evaluation approach to increase the rigour of DSS design research, by systematically evaluating checkpoints throughout the DSS development process. The results of this study show that the proposed approach has merits in assessing the quality of the entire DSS development process. This represents a more rigorous approach than existing examples, since it is validated by the design science literature. Practically, the approach provides superior options for the evaluation and ultimate acceptance of DSS in many industries, including rural examples, where user acceptance and rigorous approaches for evaluation have been particularly problematic.

The results also highlighted some issues with the study. Conveying the meaning of concepts such as quality proved challenging, although examples were presented to the respondents to illustrate this. Further, each group used a different definition of end user. However, this did not appear to affect the results, other than in the problem complexity checkpoint, where scientists as end users were considered to be unaffected by complexity, whereas plantation managers were viewed to be strongly affected. Further limitations of this study were the number of industry participants and the use of a single DSS development case. This reflects the specialised nature of the problem domain; indeed, many DSS problem domains are highly specialised.

ACKNOWLEDGEMENTS

The authors would like to thank Dr Simon Lawson, Dr Helen Nahrung, Dr Valerie Debuse and Dr Manon Griffiths for their valuable assistance with this research and in-kind support.

REFERENCES

- Arnott, D. 2006. "Cognitive biases and decision support systems development: a design science approach", *Information Systems Journal* (6), pp 55–78.
- Arnott, D. and Pervan, G. 2008. "An assessment of DSS design science using the Hevner, March, Park, and Ram guidelines", *The 4th Biennial ANU Workshop on Information Systems Foundations*, Information Systems Workshop, Australian National University, Canberra, Australia, pp 255–284.
- Arnott, D. and Pervan, G. 2005. "A critical analysis of decision support systems research", *Journal of Information Technology* (20), pp 67–87.

- Carlsson, C. and Turban, E. 2002. "DSS: directions for the next decade", *Decision Support Systems* (33), pp 105–110.
- Carlsson, S.A. 2006. "Design science research in Information Systems: a critical realist perspective", *17th Australasian Conference on Information Systems, ACIS2006*, Adelaide, Australia.
- Carnegie, A.J., Stone, C., Lawson, S.A. and Matsuki, M. 2005. "Can we grow certified eucalypt plantations in subtropical Australia? An insect pest management perspective", *New Zealand Journal of Forestry Science* (35), pp 223–245.
- Cox, P.G. 1996. "Some issues in the design of an agricultural decision support systems", *Agricultural Systems* (52), pp 355–381.
- DeLone, W.H. and McLean, E.R. 2003. "The DeLone and McLean model of Information Systems success: a ten year update", *Journal of Management Information Systems* (19:4), pp 9–30.
- El-korany, A., Rafae, A., Baraka, H., and Eid, S. (2000). "A structured testing methodology for knowledge-based systems", In: Ibrahim, M., Kung, J., Revell, N. (Eds.), DEXA 2000. Springer-Verlag, Heidelberg, pp 427–436.
- Forgionne, G. 1999. "An AHP model of DSS effectiveness", *European Journal of Information Systems* (8), pp 95–106.
- Gregor, S. and Jones, D. 2007. "The anatomy of a design theory", *Journal of the Association for Information Systems* (8:5), pp 321–335.
- Hevner, A., March, S., Park, J. and Ram, S. 2004. "Design science in Information Systems research", *MIS Quarterly* (28:1), pp 75–105.
- Hirschheim, R. and Smithson, S. (1988). "A critical analysis of information systems evaluation". In N. Bjom-Andersen & G. B. Davis (Eds.), *Information Systems Assessment: Issues and Challenges*, Noordwijkerhout, North Holland.
- Hitt, L. and Brynjolfsson, E. 1996. "Productivity, business profitability, and consumer surplus: three different measures of information technology value", *MIS Quarterly* (20:2), pp 121–142.
- Iivari, J. 2007. "A paradigmatic analysis of Information Systems as a design science", *Scandinavian Journal of Information Systems* (19:2), pp 39–64.
- Jackson, B. and Sulaksono, S. 1998. "Going soft on Information Systems evaluation", *Australasian Journal of Information Systems* (5:2), pp 41–50.
- Juristo, N. and Morant, J.L. 1998. "Common framework for the evaluation process of KBS and conventional software", *Knowledge Based Systems Journal* (11:2), pp 145–160.
- Kerr D.V. (2004) Factors Influencing the development and adoption of knowledge based decision support systems for small, owner operated rural businesses *Artificial Intelligence Review* 22(2) pp 127-147.
- Kerr, D.V. and Winklhofer, H. 2006. "The effect of rapid rural industry changes on the development of a decision support system for dairy farmers in Australia", *Computers and Electronics in Agriculture* (50:1), pp 61–69.
- Mackrell, D., Kerr, D.V. and von Hellens, L. 2009. "A qualitative case study of the adoption and use of an agricultural decision support system in the Australian cotton industry: the socio-technical view", *Decision Support Systems* (47), pp 143–153.

- Mahaman, B.D., Passam, H.C., Sideridis, A.B. and Yialouris, C.P. 2003. "DIARES-IPM: a diagnostic advisory rule-based expert system for integrated pest management in Solanaceous crop systems", *Agricultural Systems* (76), pp 1119–1135.
- Mansingh, G., Reichgelt, H. and Bryson, K.O. 2007. "CPEST: an expert system for the management of pests and diseases in the Jamaican coffee industry", *Expert systems with Applications* (32), pp 184–192.
- March, S.T. and Smith, G. 1995. "Design and natural science research on Information Technology", *Decision Support Systems* (5:4), pp 251–266.
- McCown, R.L. 2002. "Changing systems for supporting farmers' decisions: problems, paradigms, and prospects", *Agricultural Systems* (74), pp 179–220.
- McKay, J. and Marshall, P. 2007. "Science, design, and design science: seeking clarity to move design science research forward in Information Systems", *Proceedings of the 18th Australasian Conference on Information Systems*, pp 604–614.
- McKay, J., Marshall, P. and Heath, G. 2008. "An exploration of the concept of design in Information Systems", *The 4th Biennial ANU Workshop on Information Systems Foundations, ISF2008*, S. Gregor and D. Hart (Eds.), Australian National University, Canberra, Australia.
- Miah, S.J. 2008. "An ontology based design environment for rural decision support", Unpublished PhD thesis, Griffith University, Australia.
- Miah, S.J., Kerr, D. and Gammack, J. 2009. "A methodology to allow rural extension professionals to build target-specific expert systems for Australian rural business operators", *Expert Systems with Applications* (36), pp 735–744.
- Mumford, E. 2000. "A socio-technical approach to systems design", *Requirement Engineering* (5), pp 125–133.
- Muntermann, J. 2009. "Towards ubiquitous information supply for individual investors: a decision support system design", *Decision Support Systems* (47), pp 82–92.
- Parikh, M., Fazlollahi, B. and Verma, S. 2001. "The effectiveness of decisional guidance: an empirical evaluation", *Decision Sciences* (32:2), pp 303–31.
- Peffers, K., Tuunanen, T., Rothenberger, M.A. and Chatterjeea, S. 2008. "Design science research methodology for IS research", *Journal of Management Information Systems* (24:3), pp 45–77.
- Phillips-Wren, G., Hahn, E. and Forgionne, G. 2004. "A multiple criteria framework for the evaluation of decision support systems", *Omega* (32:4), pp 323–332.
- Phillips-Wren, G., Mora, M., Forgionne, G.A. and Gupta, J.N.D. 2009. "An integrative evaluation framework for intelligent decision support systems", *European Journal of Operational Research* (195), pp 642–652.
- Purao, S. and Storey, V.C. 2008. "Evaluating the adoption potential of design science efforts: the case of APSARA", *Decision Support Systems* (44), pp 369–381.