Journal of the Association for Information Systems (2019) **20**(5)**,** 570-603
doi: 10.17705/1jais.00544

RESEARCH PAPER

ISSN 1536-9323

A Metadesign Theory for Tailorable Decision Support

Shah Jahan Miah[1], John G. Gammack[2], Judy McKay[3]
[1]Victoria University, Australia, shah.miah@vu.edu.au
[2]Zayed University, United Arab Emirates, john.gammack@gmail.com
[3]Swinburne University of Technology, Australia, jmckay@swin.edu.au

Abstract

Despite years of decision support systems (DSS) research, DSS artifacts are frequently criticized for lacking practitioner relevance and for neglecting configurability and contextual dynamism. Tailoring in end-user contexts can produce relevant emergent DSS artifacts, but design theory for this is lacking. Design science research (DSR) has important implications for improving DSS uptake, but generally this has not been promoted in the form of metadesigns with design principles applicable to other DSS developments. This paper describes a metadesign theory for tailorable DSS, generated through action design research studies in different primary industries. Design knowledge from a DSS developed in an agricultural domain was distilled and generalized into a design theory comprising: (1) a general solution concept (metadesign), and (2) five hypothesized design principles. These were then instantiated via a second development in which the metadesign and design principles were applied in a different domain (forestry) to produce a successful DSS, thus testing the metadesign and validating the design principles. In addition to contributing to DSR and illustrating innovation in tailorable technology, the paper demonstrates the utility of action design research to support theory development in DSS design.

Keywords: Metadesign, Design principles, Design Science Research, Decision Support Systems, Action Design Research, Tailorable Design, Instantiation Validity

Jan Pries-Heje was the accepting senior editor. This research article was submitted on November 4, 2017, and went through one round of revision.

1 Introduction

Decision support systems (DSS) is a well-recognized tradition in information systems (IS) research (Power, Sharda & Burstein, 2015). However, DSS research and DSS artifacts often lack practitioner relevance (Arnott, 2006; Arnott & Pervan, 2012; Miah, Kerr & von-Hellens, 2014), and DSS themselves are criticized both for a lack of configurability, and also for lacking contextual sensitivity (Brézillon and Pomerol (1998); Smirnov et al, 2015). This is particularly evident in science-based domains (e.g., the agricultural sector), where the lack of DSS uptake is commonly attributed to differences between the contexts, domain-specific languages, and practices of domain experts (agricultural scientists) on the one hand, and those of domain practitioners (farmers) on the other (McCown, 2002; Carberry et al., 2002; Meensel, Lauwers, Kempen, Dessein, & Huylenbroeck, 2012).

Practitioners often judge DSS jointly designed by domain experts and DSS designers as not useful and/or too complex because of unfamiliar scientific terminology and logic (Cox, 1996; Walker, 2002). Furthermore, practitioners often regard DSS as unreliable and inflexible and as being generally unable to take account of local and changing environmental and business conditions that impact the preset parameters embedded in the DSS (Gillard & Johnson, 2001; Karmakar et al., 2007; Meensel et al., 2012; Miah et al., 2014). Participation of practitioners in design, however, has been shown to improve results, since this causes research activities and outcomes to be better aligned with participant expectations (Carberry et al., 2002; Meensel et al., 2012).

In developing relevant DSS, designing effective collaboration between DSS designer, agricultural scientist (domain expert) and farmer (practitioner) is essential to leverage the knowledge and practices of both scientist and practitioner. Hence, separating the principles of DSS design from the specific knowledge of domain participants allows conceptualization of a DSS design and development environment in which participants can contribute to develop a detailed design that is both rigorous and relevant.

Design science research (DSR) has been seen as potentially improving both the rigor and relevance of DSS research: Arnott and Pervan (2012), however, found that despite DSR's emergence in DSS research, to date there had been little improvement in the perceived relevance and uptake of DSS among practitioners. Nonetheless, in identifying relevant knowledge for designing DSS solutions, design theory as an output of DSR has potentially important applications for improving DSS design. This, however, has not generally been promulgated either in the form of generic metadesigns nor as design principles applicable to new instances of a specific class of DSS problems.

In any dynamic context, a DSS must be configurable postimplementation by practitioners if it is to be useful, as practitioner decision-making responds to seasonal and regional variability in environmental conditions and market contexts. Practitioners, the end users of the designed DSS, "ultimately…know their context best" (Bell, 1992, p. 51), and are contextually positioned to customize designs in their role as secondary designers: their role should not be seen as mere information sources to a third party, since involving more empowered practitioners in tailoring system features within the context of use helps increase the effective use of a technology (Germonprez, Hovorka, & Gal, 2011; Germonprez, Hovorka & Collopy, 2007). Design approaches built around the practitioners' contextual knowledge and normal practices results in greater inclusion in DSS design and leads to improved information quality, end-user acceptance, productivity, and performance (Miah, Kerr, & von-Hellens, 2014). Ensuring increased uptake and ongoing relevance as situations change requires a metadesign[1] such that a specific artifact is able to emerge and evolve, incorporating both specialized scientific knowledge from domain experts and contextual, practice-based knowledge from practitioners. It is thus appropriate to theorize, design, and build DSS artifacts that incorporate deep practitioner knowledge of context, and that support tailorability, empowering practitioners to respond to dynamic contextual requirements.

Gregor and Hevner (2013) describe how a design theory (an abstract, coherent body of prescriptive knowledge that explains principles of form and function, methods, and justificatory theory used to develop a design artifact) "provides a useful generalization for extending knowledge in the problem or solution domains" (p. 352). Our aim is to specify a design theory for a class of artifacts designed for decision support in the form of a generic DSS design environment and associated design principles, incorporating precepts of practitioner involvement in co- and secondary design throughout, and offering greater control in end-user contexts. We embed principles of configurability (tailorability) required for artifact mutability (Gregor & Iivari, 2006) so that the design outcome better reflects the practitioners' reality in which the DSS will be adopted and used.

Markus, Majchrzak, and Gasser (2002) propose a design theory for systems that requires emergent knowledge processing in a problem space where users face highly unpredictable sequences of activities in their work contexts. Markus et al.'s (2002) study suggests principles for a class of problems encompassing different processes, user requirements, and knowledge requirements, and noted a need both for theoretical validation and for development tools that could support integration of expert and local knowledge. Our work aims to extend DSS design theory and practice by developing a theoretical approach that generalizes across a class of domains from a DSR perspective.

Methodologically, we position our work against the two strategies of DSR distinguished by Iivari (2015), where the DSR focus is on relevant and general

---

[1] Koehne, Redmiles, and Fischer (2011) described metadesign as a theoretical framework supporting any systems that allow end users to become designers in dynamic use contexts. In this paper, the framework that we propose is grounded in the objectives such that (1) the metadesign theory can inform the essential design components relevant to contextual situations, and (2) a set of principles of design provides guidelines for further similar DSS design.

problem-solving through emergent systems beyond simply constructing a practice-informed but essentially conceptual artifact development. We utilize action design research (ADR) (Sein, Henfridsson, Purao, Rossi & Lindgren, 2011) and reference two empirical DSS case studies that were developed sequentially in specific industry client contexts and that show the development and validation of generic design principles.

The paper is based on the qualitative secondary analysis (QSA) of this sequential development of two practitioner-oriented DSS development projects. The first was in the dairy industry (Miah, Kerr, & Gammack, 2009; Miah, 2008); the second in forestry pest management (Miah, Debuse, Kerr, & Debuse, 2010). Previous publications mainly describe the DSS solution design and the underpinning methodology used to address specific industry decision support issues. The first design case was further elaborated in a DSR perspective on DSS artifact development more deeply in that it considered evaluation (Miah et al., 2014). To extend the value of these works it is important to conduct a secondary analysis comparing and contrasting the key findings from both design cases to generate new generalizable understanding and knowledge.

Secondary analysis reuses data to glean new understandings, and though qualitative secondary analysis is relatively recent, it is growing as a methodological approach as data sets become more widely available online (Tarrant, 2016). Several studies in IS have used secondary data in quantitative studies—for example, da Costa Campos (2015) used data from past Facebook advertising campaigns to identify the effects of user-generated content, and Ryder (2005) used census data to investigate the digital divide on the Isle of Man. Qualitative secondary analysis is rarer, but as Arnott, Lizama and Song (2017) point out in the context of their own analysis of eight business intelligence systems, increases in data lead to greater generalizability, and the data are likely to be of higher quality when the original researchers are involved. Arnott et al. (2017) suggest that this is due to the original researchers' deep understanding of the data's meaning and further argue that fit between available data and secondary analysis requirement is ensured when similarities in phenomena studied, data collection, and unit of analysis apply. As such, for the current paper we conducted a secondary qualitative analysis across the case findings to assess how the approach taken in the first case (dairy industry case) might generalize or adapt to the second case (forestry pest management case). We aimed first to define a set of metarequirements and through the application of DSR strategies (Iivari, 2015) developed a metadesign in the first case, which was then instantiated and evaluated via the second case. This enabled us to develop a metadesign framework and design principles to construct a new DSR theory to inform and improve DSS design practices for a particular class of problems: an issue we will explore further in discussion.

The rest of the paper is structured as follows. The next section discusses relevant background literature in DSS design, elaborating the issue of practitioner relevance. The next section describes our research approach, which uses Iivari's (2015) two strategies for DSR to scaffold a design process within an action design research framework (Sein et al., 2011). The section after that provides details of the design theory generated by our research, which is followed by a discussion section that critiques our contribution—namely, a meta-artifact design and related design principles for an identified problem class. The paper concludes by acknowledging the limitations of the research and considering future research aimed at further extending DSR knowledge for DSS design.

2 Literature Review

2.1 DSS Design Issues

DSS has been a prominent research field in IS for the last four decades (Hosack, Hall, Paradice & Courtney, 2012). There are, however, repeated claims that DSS—particularly those intended for individual decision makers (personal DSS)—lack both configurability and contextual sensitivity and are thus generally not responsive to changing conditions and environments (Arnott & Pervan, 2008; Meensel et al., 2012). Individual decision-making preferences and the cognitive styles of practitioners are also generally overlooked by DSS designers (Arnott, 2006). DSS research is thus largely ignored by practitioners, as it is deemed to be irrelevant to meet their needs (Arnott & Pervan, 2008; Vizecky & El-Gayar, 2011). While acknowledging this challenge, research that explicitly pays attention to the issue of practitioner relevance has been limited (Arnott & Pervan, 2014). This is particularly the case in the design of DSS that require specialized knowledge from domain experts to support the development of appropriate algorithms and parameters within the DSS, but where changing contextual and practice-based knowledge capture may also be vital to acceptance (McCown, 2002; Carberry et al., 2002).

Table 1: Factors Contributing to a Lack of Practitioner Relevance in Primary Production DS

Broad themes	Agricultural DSS design issues	Sources
Lack of practitioner knowledge capture and engagement	Gap in co-knowledge production among practitioners and DSS designers	McCown, 2002; Hayman & Easdown, 2002; Cox, 1996
	Differences between scientific knowledge and practice-based knowledge; scientific terminology and logic that is unfamiliar to practitioners	Meensel et al., 2012;McCown, 2002; Carberry et al., 2002; Miah et al., 2014
	DSS development methods offer few opportunities for practitioner input	Karmakar et al, 2007; Gillard & Johnson, 2001; Kerr et al.,1999
	Inflexible update options in DSS	Valls-Donderis et al., 2014; Churi et al., 2013
Lack of tailorability and contextual sensitivity	Lack of configurability	Meensel et al., 2012; Karmakar et al., 2007; Gillard & Johnson, 2001
	Lack of contextual sensitivity	Brézillon and Pomerol, 1998; Cox, 1996; Miah et al., 2014
	Failure to cater to changing requirements caused by environment change, industry change	Kerr & Winklhofer, 2005; McCown, 2002; Cox, 1996
	Too static and too complex; unreliable and inflexible provisions available for practitioner users	Lambert & Elix, 2003; Walker, 2002; Cox, 1996

Table 1 illustrates that the issues of relevance can be viewed as clustering around two broad themes: (1) that current practice struggles with adequately engaging practitioners and in eliciting and representing their practice-based knowledge in the DSS, and (2) that developed DSS exhibit a lack of configurability and contextual sensitivity and fail to cater to dynamic contexts. To improve design practice, it is thus important that more attention is paid to each of these challenges.

3 Practitioner Engagement and Knowledge Transfer

A common approach reported in the DSS literature has been for the DSS designer to work closely with the domain expert (scientist) whose knowledge is then captured within the resultant DSS design. This is particularly evident in DSS developed in the agricultural sector, where scientific knowledge of agricultural scientists is embedded in the DSS with the aim of supporting decision-making of farming practitioners in the field (Voinov & Gaddis 2008; Walker 2002). However, this tends to result in the problems identified above that are associated with the failure to recognize that practitioner experience and knowledge of context may also need to be included in any system designed to support practitioner decision-making. The domain expert may lack the practitioner's knowledge of context, localized practices, and circumstantial variabilities—which, if ignored, results in a system that is perceived by the end user (the practitioner) as lacking utility.

Van de Ven and Johnson (2006, p. 806) distinguish two knowledge directions that operate in different contexts and serve different purposes:

> *the purpose of practical knowledge is knowing how to deal with the specific situations encountered in a particular case. The purpose of scientific knowledge is knowing how to see specific situations as instances of a more general case that can be used to explain how what is done works or can be understood.*

Achieving success with DSS endeavors is thus problematic and involves the challenge of bridging two types of knowledge—the research-driven theoretical knowledge of the scientists with the contextual, practice-based knowledge of the farmers, the ultimate artifact users.

Carlile (2004) and Edwards (2012) provide further theoretical insights into the issues of common knowledge and combining knowledge across professional boundaries. Transferring knowledge across science-based and practice-based professional boundaries need not involve fully understanding the work and knowledge of others, but rather recognizing and respecting what others know and translating based on a common understanding (Edwards, 2012). In practice, we followed Boland and Tenkasi (1995) who examine collaboration across diverse communities of knowing, recognizing two distinct critical processes: (1) *perspective making* is the process of knowledge creation and sharing within a community that serves to strengthen and complicate the problem-solving capabilities of the group, and (2) *perspective taking* is the process through which members of one community come to appreciate, integrate, and use the knowledge of a different community. Clearly, the success of the DSS rests on perspective taking: all parties need sufficient mutual appreciation to accommodate their diverse knowledge within the artifact design in order to make a stronger unified perspective. Effective communication thus relies on all relevant parties understanding the language games (Wittgenstein, 1953, 2001) involved in a specific context and interaction. In the case of collaborative development of DSS, this is complex, involving not only the scientific jargon of the domain experts and the practical know-how of the practitioners, but also the technical, design, and process knowledge of the developer: a tripartite arrangement of knowledge sharing and translation. The success of the DSS rests on all parties achieving sufficient mutual appreciation to accommodate their diverse knowledge within the artifact design. Technologies such as controlled vocabularies and ontologies[2] that support effective knowledge transfer and knowledge translation thus become critical to the success of DSS designed to address this type of problem.

Kayande et al. (2009) provide theoretical support for this in a DSS context arguing that the decision model embedded in the DSS must align with the mental model of the user. Proposing a framework designed for domains with repetitive decisions and uncertain outcomes, they argue that acceptance is enhanced when users can understand the rationale for a decision in familiar terms. They identify potential gaps between the DSS model, the user's model, and the "true" model and explore the role of feedback in reducing the gaps between the DSS and the true model (the smaller the better) and between the DSS and the user's mental model, such that users can learn and accept objectively accurate DSS. We implement the spirit of this idea by ensuring that the science informed the DSS model is accommodated, but also by making sure that user vocabulary, knowledge and ideas are accommodated as well. The basic DSS model can be tailored by users with feedback on the success of this evident to all stakeholders—potentially helping the user learn a truer model and even reciprocally contributing to scientific knowledge.

3.1 Tailorable Technology Design

The second broad theme contributing to the perceived lack of relevance of DSS is tailorability. Inherent in the notion of tailorable technology design is a view that design needs to evolve from ready-made "packaged" technology to technology that affords greater opportunity for users to interpret and re-create the technology according to their own needs and contexts (Gasson, 2003; Germonprez et al., 2007). DSS often lack the ability for end users to customize features according to their contextual circumstances, suggesting a need for a human-centered perspective of DSS design (Gill, 1996) that views system users as secondary designers—i.e., "active, aware, and intentional participants in an ongoing process of embodied interactions involving technological and social dualities" (Germonprez et al., 2011, p. 663).

Secondary design is an activity that is underrecognized in design theorizing (Hovorka, 2010, p. 20) and is conceptualized as an ongoing activity for end-user tailoring of IS applications to maintain fit within active contexts. Tailorable design essentially requires that a technology contain dynamic, recognizable components and conventions for enabling users to intentionally modify IS features to better suit their objectives and requirements, and thus improve the utility of the system. This has affinities with the Scandinavian tradition of user-involved design, theorized by, (among others), Friis (1996). Therefore, a dual-design perspective is adopted here, recognizing the primary design activity of a pre-use artifact-building environment (led by the DSS designer), and potentially multiple instances of the used artifact, with users acting as secondary designers.

3.2 DSR for Tailorable DSS Design

While there is broad acceptance of the value of evolutionary development for decision support projects, there is little advice available to DSS developers about how to proceed with evolutionary activities (Arnott, 2006) or guidance on secondary design (Hovorka, 2010). In most cases, the objective of DSS projects is to improve the decision process and outcomes of managerial decision-making. The DSS developer needs to have a clear idea of the nature of the target decision task and a clear strategy of how to support the decision process. DSS developers, however, tend to focus on the improvement of decision outcomes through iterations of the development process, rather than through focusing on the practitioner's context and the nature of the problems faced (Arnott & Pervan, 2010). Artifact design knowledge that is the outcome of DSR offers little specific guidance into how a designer might incorporate contextual issues into their DSS designs, as the current orthodoxy in IT design science explicitly excludes contextual requirements (Carlsson, 2007).

---

[2] An ontology is commonly understood to be "a formal, explicit specification of a shared conceptualization" (Studer, Benjamins, & Fensel, 1998), and all the elements of this definition are relevant to knowledge-based systems such as DSS.

Hevner, March, Park, and Ram (2004) articulate a problem that DSR researchers face: "the existing knowledge base is often insufficient for design purposes and designers must rely on intuition, experience, and trial-and-error methods" (p. 99). A similar problem confronts DSS designers: the current knowledge of DSS design approaches is inadequate for "real world" DSS design contexts where secondary design activities are required to improve the relevance of decision support outcomes. Given the nature of industry-specific DSS involving both domain experts and practitioners, and drawing on both scientific and practice knowledge, some of the need for DSS designers to draw on intuition and trial and error approaches may be reduced. This is because of the structured nature and ready availability of much of the scientific knowledge in agriculture. This claim is further supported by Arnott (2006), who argued that because DSS is (in the end) about decision-making, a DSS designer should access considerable practice-oriented knowledge about client-specific decision processes. In so doing, the DSS designer would therefore have to rely less on intuition and trial and error. The term "access" suggests a need for DSS designers to appreciate and design for (rather than attempt an exhaustive replication of) practitioner-centered realities—including their subjective judgments and decision processes—in order to effectively provide relevant support.

However, many of the construction-centric approaches that dominate DSR (McKay, Marshall & Hirschheim, 2012; Iivari, 2015), and DSS development more generally, arguably prioritize rigor over relevance. These approaches focus primarily on the development of artifacts addressing a general problem, which may or may not be applied in practice. These are not developed for a specific client and the practical relevance of this strategy (which Iivari (2015) calls Strategy 1) "varies greatly" (Iivari, 2015, p. 110). The lack of relevance to, and application in, specific client contexts contrasts with "Strategy 2" in which a specific client problem drives a real implementation, which is "a priori better equipped to address immediate practical problems" (Iivari, 2015, p. 110).

Moreover, as practitioner contexts are frequently quite volatile, tailorability becomes an important requisite to sustaining a relevant and useful support mechanism that empowers the practitioner to modify, configure, and redesign the DSS artifact, since they respond to changing contextual features. In the context of DSS, this suggests a need to extend development to provide designs relevant to complex, sociotechnical contexts (Miah et al. 2014; Miah et al. 2009), which involves not just building an artifact that works and that is tailorable during ongoing use, but one that includes domain experts and practitioners throughout the design process and one that recognizes the influence of the contextual characteristics in which that artifact is deployed. In addition to being foreshadowed by Keen (1980), who states "the final system must emerge through an adaptive process of design and usage" (p. 9), this is espoused in Sein et al. (2011) who, along with Markus et al. (2002), exemplify Iivari's Strategy 2 and emphasize emergent artifacts.

DSS designers thus need to move beyond traditional DSS architectures featuring the interrelationships of the major essential DSS design elements (Holsapple, 2008) as their source of knowledge about design. Even DSS generators, generally based on traditional DSS architectures within a package of related integrated software that provides a set of capabilities to quickly and easily develop a specific DSS (Power, 2002), are likewise of limited use for designers keen to overcome the lack of relevance challenges associated with DSS. These are typically spreadsheet or traditional application development environments, not intended for practitioner end users. In DSS, there are no "full service" DSS generators, so creating effective DSS development environments remains important for meeting the expectations of clients (Hosack et al. 2012; Power, 2004). In contexts where scientific and practitioner knowledge need to be included in the DSS and provisions made to empower secondary design activity, we propose that a differently architected DSS design environment is required to meet contextual decision support needs.

Winograd (1995) argues that software engineering requires a shift from programming environments to design environments to better to satisfy end users' cognitive needs and to help deal with contextual issues (social, cultural, and aesthetic) that impact users and software applications (Gammack, 1999). This idea has been since applied and extended in other domains—e.g., for empowering creative knowledge work (Fischer, 1999), for wiki design in teaching IS (Kane & Fichman, 2009), and in online communities (Ren, Kraut, & Kiesler, 2007). We adopt the design environment terminology specifically, as we argue that DSS designers cannot focus simply on the technical system but must also take into account the context-of-use requirements of end users. In terms of adaptive design and use, this is in line with Keen's (1980) definition of DSS, requiring theoretical attention to the dynamics between user, designer, and system. A DSS design environment reminds DSS designers of the need to center decision-making with the end user, and to allow configurability of the DSS to cater to specific, localized variability of context where decisions are enacted.

Despite an early proposal by Sol (1987, p. 11) to "direct DSS-research to the concept of DSS-generators or, more generally, DSS-design environments" there remain few, if any, DSS design environments specified as design science contributions. This has effectively consigned much DSS work to the category of specific development projects, or underused conceptualizations, rather than producing generalizable principles that provide more enduring design knowledge. Design science research, however, explicitly seeks to identify the knowledge contribution of an artifact development (Gregor & Hevner, 2013) to increase the field's knowledge base.

From a research perspective, DSR knowledge is essential to understanding the requirements for such a DSS design environment, and in design theory development, action research (in which hypothesized principles are evaluated and iterated) is appropriate (Walls, Widmeyer, & El Sawy, 1992; Markus et al., 2002). Walls et al. (1992) describe two types of principles, some governing the design or selection of features of a system, with others addressing the development process. Markus et al. (2002) combined these in generating six theoretical principles for emerging knowledge processing systems design, providing an important contribution to IS design theory. In the present research, knowledge would be generated as a result of iterating design activity, resulting in the articulation of a generic DSS design environment. Theoretically, this will include a set of metarequirements, an instantiated meta-artifact, and associated design principles applicable to an identified class of problem. Specifically, the objective of this research is to demonstrate how meta-artifact design knowledge and relevant design principles can be evolved from client-specific artifact design involving knowledge transfer, translation, and tailorability.

Specifying design knowledge that is encapsulated within the theorized DSS design environment enhances design applicability and generalizability to other client-specific artifact designs across analogous industry sectors. We frame our work in terms of Iivari's (2015) strategies, developing the DSS solution for decision support around concrete client problems and abstracting principles that generalize to a class of problem—which are also validated through applying and evaluating these principles to an instantiation in a different client context. This is detailed in the next section.

4 Research Approach

The research reported here uses qualitative secondary analysis to reanalyze and extend the authors' previous work (Miah et al. 2009; Miah, 2009; and Miah et al. 2010) involving two case studies of DSS design that focused on client-specific problem-solving and which contributed primarily to the DSS literature. These cases each involve primary industries: namely, dairy farming and forestry. Iivari's (2015) "Strategy 2" accommodates emergent systems that are more likely to be of immediate practical relevance, but it also identifies and generalizes design lessons into a general solution concept applicable to an identified class of problems. Here, we consider the two case studies from the perspective of their contribution to DSR, by articulating the meta-artifact design and relevant design principles distilled from the client-specific artifact designs. Using QSA across cases increases the research scale and the "empirical quantum", which leads to greater generalizability (Arnott et al., 2017, p. 61), while adopting action research principles throughout the research design to leverage and refine learning from earlier cycles allows theory building beyond immediate empirical contexts. This approach fits with the form of QSA known as analytic expansion (Thorne, 1994), in which primary data sets are used to address questions deriving from the previous analysis but not specified in the original research. This transition from specific to general also addresses Principle 7 in Sein et al.'s (2011) action design research (ADR) method, such that both the problem and solution instances are generalized and design principles are derived from the research outcomes.

The dairy case was conceptualized and developed as a generic design environment, from which specific DSS could be (and were) built by end-user practitioners (Miah et al. 2010): the forestry case illustrates how the design principles formalized from this could be applied and adapted to a different industry. The overall approach taken is detailed in Figure 1.

Specifically, in the first case involving the Dairy DSS, the design focus was on articulating a viable decision support solution for industry-specific decision problem-solving. Hevner et al.'s (2004) DSR framework provided methodological support, capturing problem details and user decision-making requirements for a flexible DSS artifact design, developed through the use of evolutionary prototyping that ensured the active participation of stakeholders (Appendix A details how the framework was used). In this first design case, it was found that Hevner et al.'s (2004) framework gave limited guidance on domain knowledge acquisition, of clear importance to any DSS artifact design (Kersten et al. 2002; Sowunmi et al. 1996; Arnott, 2006).

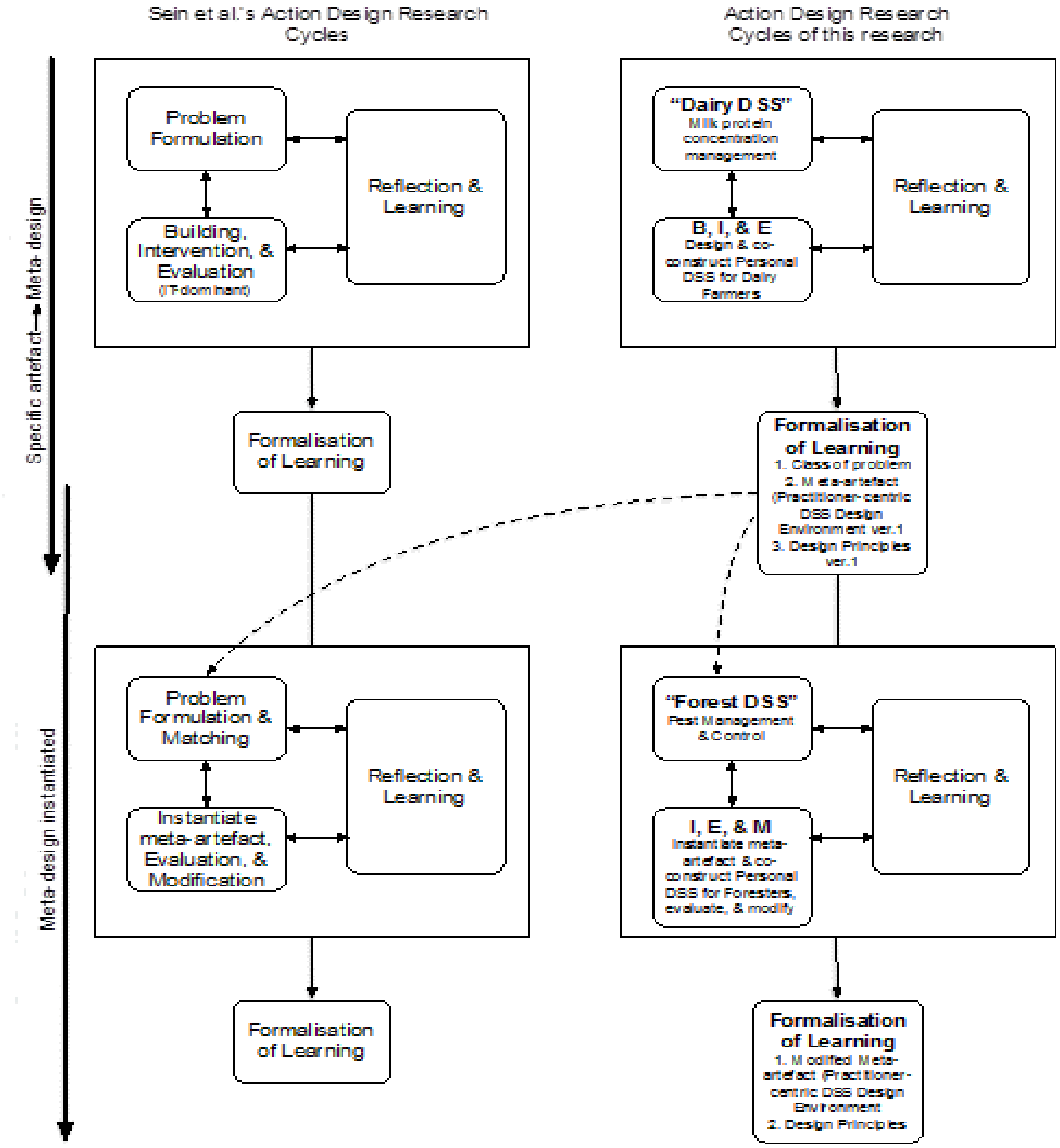


Figure 1: Our ADR-Based Research Design (Sein et al. 2011)

In this case, it was important to accumulate scientific knowledge from literature and experts, along with practice-based heuristics identified through focus groups. The acquired knowledge was then used to develop rules and represent the decomposed components via a top-down decision support approach. This activity was mainly intended to create decision rules and parameters and support alternative options for representation (such as constraints); the framework was broad enough to capture this particular need.

Appendix B illustrates how this framework was adapted to the second case. The DSR methodology of Purao and Storey (2008), developed for DSS reuse-based design purposes, proposed "to evaluate design research outcomes, when the artifact cannot be immediately deployed in an organizational setting" (p. 372). Our cases appropriately fitted their DSR framework therefore this was adopted in the second case: designing the "Forest DSS".

In both cases, our studies focused on participatory artifact development using design science methods as a solution to articulated practical decision-making issues. While not originally specified as ADR studies, both cases involved iterative participatory development cycles of build and evaluation with continual interventions following initial problem formulation. Looking toward the broader class of problems, and ultimately the formalization of design principles, in this paper we revisit the cases, describing the process in ADR terms in order to abstract a new metadesign theory for supporting DSS designers in similar domains.

It is important to consider what "similar domains" means in order to identify the theory's scope of applicability. The design environment artifact was intended to be generic across domains, such that a specific DSS could be secondarily designed and tailored in an operational context, while adhering to the built-in domain knowledge. Both our cases here were primary-industry cases, with scientifically established parameters and settings, which also lent themselves to rule-based advice given particular input values.[3] With experts providing the knowledge in two relatively well-structured domains, the recommendations have a high probability of being optimal, and, by including user representatives throughout the development, of being both understood and relevant.

With these two cases, we were able to effectively apply Iivari's (2015) two strategies for DSR. The first strategy (Strategy 1) is evident when DSR researchers first construct an IT metadesign as a general solution concept, which can potentially be instantiated in multiple, specific solution contexts. Strategy 2 is apparent when researchers attempt to address a client-specific problem by building an IT artifact for that specific problem context, and then distill from it knowledge that can be generalized into a metadesign. The two empirical design cases on which this paper is based illustrate a Strategy 2 approach, that enables learning about meta-artifact design and suggests initial design principles. This approach is followed by a Strategy 1 approach to the second case (albeit using a Strategy 2 development), through which the emerging general solution concept in the form of metadesign and design principles were tested, evaluated, and further refined in a different instantiation.

These strategies are conceptually implemented here using an ADR framework (Sein et al., 2011) (Figure 1). ADR is based on the premise that IT artifacts both shape and are shaped by the organizational context in which they are implemented, and should be regarded as "ensemble artifacts…result[ing] from the interaction of design efforts and contextual factors throughout the design process" (Sein et al., 2011, p. 38). In each cycle, the research team was mindful of the need for close collaboration with stakeholders, joint approaches to problem-solving, and co-design of the evolving DSS. Details of the participants in the two studies are given in Table 2.

Table 2: Participants Details of Two Design Cases

Design Cases	List of participant and details
Dairy DSS	**Extension officers (domain experts):** Extension officers are agricultural scientists employed by Queensland Department of Primary Industries to provide business support services to dairy farmers to help them, for example, enhance their milk production by applying different and new scientific methods to their farming practice. Three extension officers were involved in this study (DE-d1-3). Another three specialized dairy experts, a dairy physiologist, a dairy nutritionist, and a dairy breed specialist were involved to provide expert inputs (DE-d4-6). **Farmers (practitioners):** These were the business owners and managers who conduct day-to-day business activities around dairy herd management, including breeding, managing feed requirements, milk production monitoring, and so on. Seven practitioners (P-d1-6) were involved in this study.
Forestry DSS	**Forest scientists (domain experts):** Forestry health scientists in the Department of Employment, Economic Development & Innovation, Queensland are involved in research and analysis of plant improvement, insects, and diseases in order to minimize pest and disease impacts on forestry growth. Scientists provide advice and work alongside foresters and plantation managers to ensure that the latest research and development is directed toward helping reduce the damage to forest and plantation growth and sustainability. Two domain experts were involved in this study (DE-f1, De-f2). **Foresters (practitioners):** Foresters or plantation managers manage forested lands and care for trees; supervise other workers in determining the type, number, and placement of trees; and assist in identifying insects capable of damaging the trees. They also manage tree nurseries, pests, diseases, monitor growth, and assess sustainability. Two practitioners (P-f1, P-f2) were involved in this study.

---

[3] It is worth noting that, at this level of abstraction, a tailorable system to build various finance and insurance DSS applications was independently developed by one of the authors based on bank record big data, augmented by experts to reflect new business rules or policies, and adjustable to local contexts by end users (Gammack et al, 1992), suggesting generality to a dissimilar industry sector.

4.1 ADR Cycle 1: DSS for Dairy Farmers

This section describes the design and development process in detail for the first ADR cycle (the "Dairy DSS").

4.1.1 Problem Formulation

In response to Australia's dairy industry deregulation, the Queensland Department of Primary Industry (QDPI) established a research project aimed at improving dairy farmers' decision-making around the management of milk protein concentration and dairy herd management more generally. Meeting industry standards for milk protein concentration throughout the year is particularly challenging in Queensland due to complicating features of the climatic environment (temperature, humidity, rainfall, etc.) and their impact on lactating animals and forage quality. Milk protein concentration impacts on milk volume pricing and the view of QDPI was that scientific knowledge encapsulated in a DSS could improve the practices of dairy farmers and, thus, the economics of the industry. An initial system development involving cooperation between scientists in QDPI and a DSS designer failed: farmers flatly refused to use the DSS complaining that it was impractical and did not account for many factors that they had to deal with daily. They argued it was too static to be useful and could not be configured to suit their individual businesses. Furthermore, they found the scientific language difficult to understand and different from the language they themselves employed. The farmers did, however, agree that the DSS concept had potential to support their decision-making in a number of key areas. QDPI recognized that a different approach was required and sought our involvement.

Investigation of the DSS failure surfaced design issues: only the scientific knowledge of the domain experts (agricultural scientists) had been included in the knowledge base; the practitioners (dairy farmers) had not been involved in the system design, nor could they tailor the implemented DSS in response to local conditions. Queensland, at 2.5 times the size of Texas, is one of the largest states in Australia and covers various climate zones, but the DSS had been designed and built without reference to the contexts in which it was to be embedded. Three persistent themes emerged: the DSS needed to be comprehensible to practitioners, it needed to meet their practical concerns, and it needed to be configurable to cater to environmental variations.

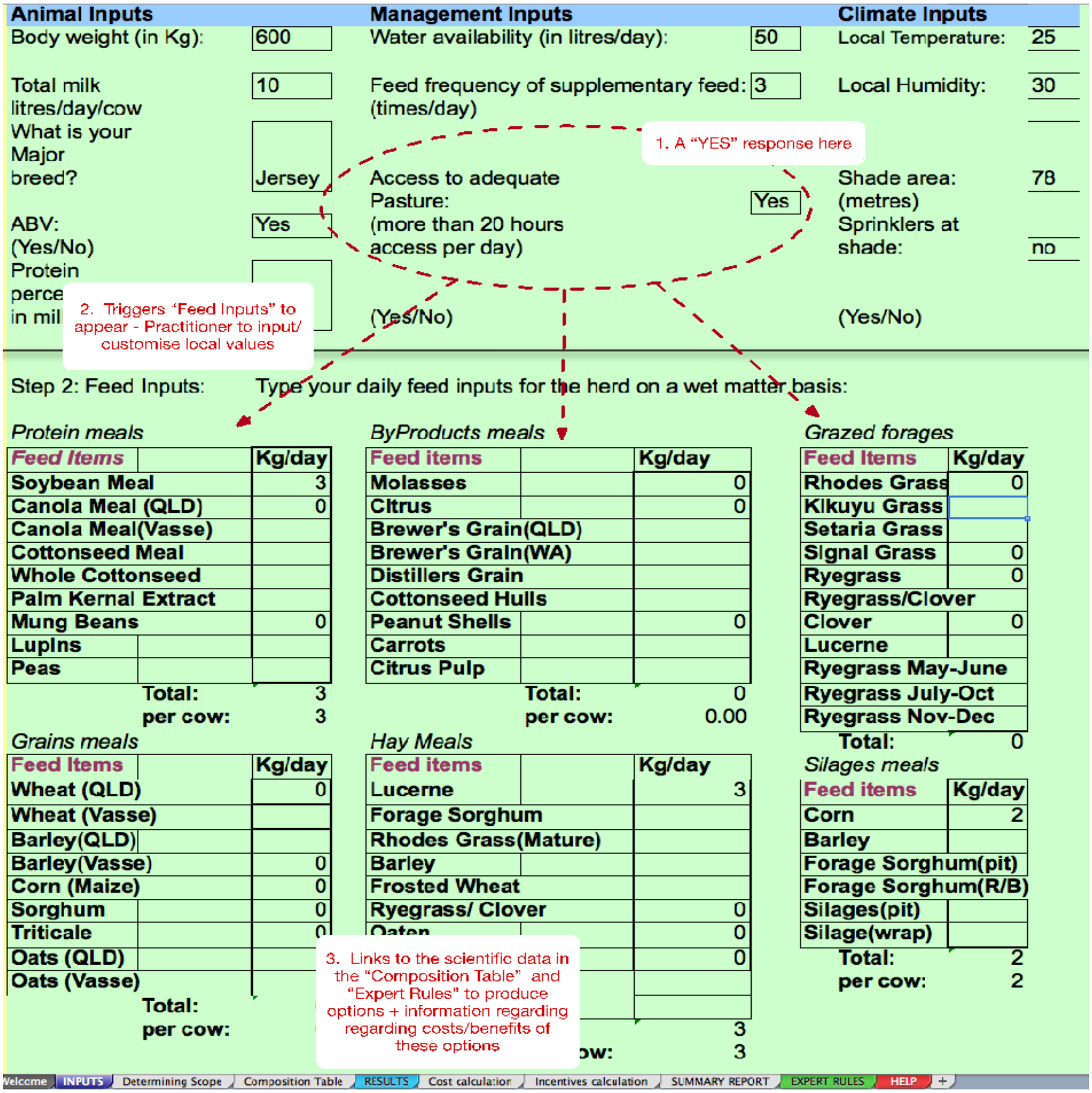
4.1.2 Building, Intervention, and Evaluation

These concerns informed our approach to designing a new DSS. Six experienced dairy farmers and six domain experts (three QDPI extension officers, a dairy nutritionist, a dairy physiologist, and a dairy breed specialist), were closely involved in collaborative development with the DSS designer (one of the researchers) throughout iterative stages of designing, prototyping, implementing, and evaluation.

The DSS designer initiated the preliminary design of the artifact but worked closely with domain experts and practitioners on melding and systematizing the experts' scientific knowledge with the practitioners' context-specific knowledge and its appropriate representation within the DSS. Through numerous interviews, project meetings, and facilitated focus groups, these diverse knowledge types became appreciated and represented within the DSS. A generic, proof-of-concept prototype (in MS Excel) was iteratively constructed and tested by both the domain experts and the practitioners, before the final DSS was built in a .NET environment. Domain experts and practitioners "played" with the evolving prototype and their feedback and suggestions were incorporated into subsequent iterations.

One key requirement was to allow for configurability and customizability in response to localized variability, resulting in multiple unique instances of the artifact in use, while ensuring the integrity of the overall system. The need for tailorability is illustrated by considering the impact of grass quality on milk protein concentration. Well-established causal models explain the seasonal impacts (temperature, rainfall, and humidity) on the moisture and fiber content in grasses (Chamberlain, 2006), directly impacting nutritional differences in milk protein concentration. Practitioners noted that they needed to reconfigure the system to account for unseasonal rainfall levels, their specific local grass types, and conditions, but also noted that the impacts of nutritional differences varied according to the cow's stage of lactation. For each set of additional or changed parameters input by the practitioner, they could save a specific, tailored version of the DSS as a benchmark should similar conditions arise in the future. Figure 2 indicates tailoring by the practitioner to account for local variable conditions. By responding "Yes" to "Access to adequate pasture", the system would reveal appropriate item description lists, enabling practitioners to input current practices. These inputs were then linked to a composition table containing relevant scientific knowledge. Advice would then appear on the screen for practitioners, informing them of the adequacy of their current practice and providing cost-benefit analyses of current and other feed mix options to improve milk protein concentration.

Through this process, the domain experts (DEs) came to appreciate the practical knowledge and contextualization that was required to produce an artifact that would really support the practitioners.



Animal Inputs

Body weight (in Kg):	600
Total milk litres/day/cow	10
What is your Major breed?	Jersey
ABV: (Yes/No)	Yes
Protein perce in mil	

Management Inputs

Water availability (in litres/day):	50
Feed frequency of supplementary feed: (times/day)	3
Access to adequate Pasture: (more than 20 hours access per day) (Yes/No)	Yes

Climate Inputs

Local Temperature:	25
Local Humidity:	30
Shade area: (metres)	78
Sprinklers at shade: (Yes/No)	no

1. A "YES" response here

2. Triggers 'Feed Inputs" to appear - Practitioner to input/ customise local values

Step 2: Feed Inputs: Type your daily feed inputs for the herd on a wet matter basis:

Protein meals

Feed Items	Kg/day
Soybean Meal	3
Canola Meal (QLD)	0
Canola Meal(Vasse)	
Cottonseed Meal	
Whole Cottonseed	
Palm Kernal Extract	
Mung Beans	0
Lupins	
Peas	
Total:	3
per cow:	3

ByProducts meals

Feed items	Kg/day
Molasses	0
Citrus	0
Brewer's Grain(QLD)	
Brewer's Grain(WA)	
Distillers Grain	
Cottonseed Hulls	
Peanut Shells	0
Carrots	
Citrus Pulp	
Total:	0
per cow:	0.00

Grazed forages

Feed Items	Kg/day
Rhodes Grass	0
Kikuyu Grass	
Setaria Grass	
Signal Grass	0
Ryegrass	0
Ryegrass/Clover	
Clover	0
Lucerne	
Ryegrass May-June	
Ryegrass July-Oct	
Ryegrass Nov-Dec	
Total:	0

Grains meals

Feed Items	Kg/day
Wheat (QLD)	0
Wheat (Vasse)	
Barley(QLD)	
Barley(Vasse)	0
Corn (Maize)	0
Sorghum	0
Triticale	0
Oats (QLD)	
Oats (Vasse)	
Total:	
per cow:	

Hay Meals

Feed items	Kg/day
Lucerne	3
Forage Sorghum	
Rhodes Grass(Mature)	
Barley	
Frosted Wheat	
Ryegrass/ Clover	0
Oaten	0
	0
	3
ow:	3

Silages meals

Feed items	Kg/day
Corn	2
Barley	
Forage Sorghum(pit)	
Forage Sorghum(R/B)	
Silages(pit)	
Silage(wrap)	
Total:	2
per cow:	2

3. Links to the scientific data in the "Composition Table" and "Expert Rules" to produce options + information regarding regarding costs/benefits of these options

Figure 2: Configurability Function for the Dairy Farmers

For example, the practitioners knew (and this was subsequently verified scientifically) that the temperature humidity index was somewhat unreliable, depending on the amount of shade or cooling systems available for the cows. The practitioners came to appreciate how scientific knowledge about breeding, herd management, pasture management, and so on could improve productivity and business outcomes for their farms. The process continued until the DEs were satisfied that the modeling was producing scientifically reliable outcomes, and the practitioners felt comfortable that they could configure parts of the DSS according to local contextual variability and that the DSS thus provided reliable support in understanding the financial and productivity implications of decisions made based on both sound science and their own knowledge of local conditions.

At the final focus group meeting, all participants had an opportunity to provide feedback on the final DSS and the design process. Despite varying previous experience with IT and DSS, the practitioners were consistently enthusiastic and positive about their experiences. Table 3 contains a sample of the comments describing experiences with the DSS, both for product and process, grouped under categories primarily addressing usability, practical relevance, and generalizability to other applications.

Table 3: Feedback from Dairy DSS Participants

Key aspects	Sub criteria	Comments
Design product	Final DSS	"It's good, very easy to navigate, I found it quite easy to use and understand" (P-d3, inexperienced with DSS). "Excellent. Easy to add and remove parameters" (P-d6, previous DSS user, confident). "It's like a simple means of organizing your thoughts into a logical framework" (P-d6).
	Strengths	"By looking at the system, I can find many answers for better results, and estimating costs" (P-d3). "Its simplicity. Really liked getting feedback from DEs" (P-d1). "It's user-friendly, compatible with my normal computer system" (P-d2). "It has the capacity to allow you to go as deep as you wish" (P-d6). "This a system where what you know you put it into the system and view it…this gives you a better understanding what you are actually doing in the field. You might think you know but sometimes you need some sort of evidence" (P-d5). "Really helpful for farmers to monitor their own productivity and other aspects of dairy management" (DE-d2). "I can see it would be applicable to a range of other rural industries" (DE-d1). "The system seems overall simple and straightforward in [terms of] data entry to me" (DE-d2).
	Improvements needed	"Think about training other users" (P-d3). "Simple but comprehensive user manual would help others. You need to have some computer skills" (P-d2). "A few examples of how the system could be used would be helpful for others" (P-d1). "Need someone to use it constantly over 12 months to give details on what could be improved" (P-d6). "It does not handle the biological consequences of certain decisions like average body weight and lactation stage to improve the capacity of the system—they should be knowledgeable enough to do that" (DE-d2).
	Applicability to other agricultural contexts	"It's got many applications. I'm thinking about how to use this system to assess goat diseases" (P-d2). "I think this system can be used for different farming methods" (P-d3). "Definitely [for] sheep and beef" (P-d6). "Traditional farmers might balk at new modern systems" (P-d1). "Would be really helpful a start-up farmer" (DE-d3). "It would be good to incorporate biological settings but I think this would be really difficult" (DE-d2). "This system gives support for decision-making in livestock-based rural operations. I can see it could also be workable for chicken and crop-based industries" (DE-d1).
	Value	"I think it could improve my decision-making" (P-d1). "Very beneficial for dairy farmers" (P-d2). "It can make my production better" (P-d3). "It has fine-tuned my operation in dairy…it helps me see what I should be doing" (P-d4). "This DSS offers individual advantages, it means you could effectively build your own DSS" (P-d6) (referring to the ability to save specific DSS tailored to a particular context) "It provides financial estimations of the implications of various decisions" (DE-d1).
	Utility	"It's handy and useful for everyday use" (P-d3). "Able to modify this framework daily/constantly, suited to addressing a specific issue on an individual farm" (P-d6). "Easy to follow the prompts in the system" (P-d2). "Information can be passed through this system by QPDI and can be used by us farmers" (P-d3). "It's helpful for knowledgeable farmers to hone their decision-making further" (P-d6) "It can formulate the daily rations for feeding" (DE-d1).
Design process		"Yeah, it was a good process, easy to follow" (P-d3). "Easy to build a new system" (P-d1, referring to the ability to save specific DSS tailored to his particular context). "This system will be helpful because of its simplicity…data entry seems to be straightforward…The system can formulate the ration in daily feed, it can also add the financial estimation to make decisions" (DE-d4) "I think it's a great way of transferring expert knowledge in a way useful to practitioners…Extension officers [Domain Experts] can transfer scientific knowledge to the farmers and associated with this they [practitioners] can build their own decision support applications. This information could be important for farmers to improve their production as the farmers are the primary beneficiary of this system" (DE-d1).

Note: P = Practitioners ; DE = Domain experts; d (1 to N) = Dairy farmers

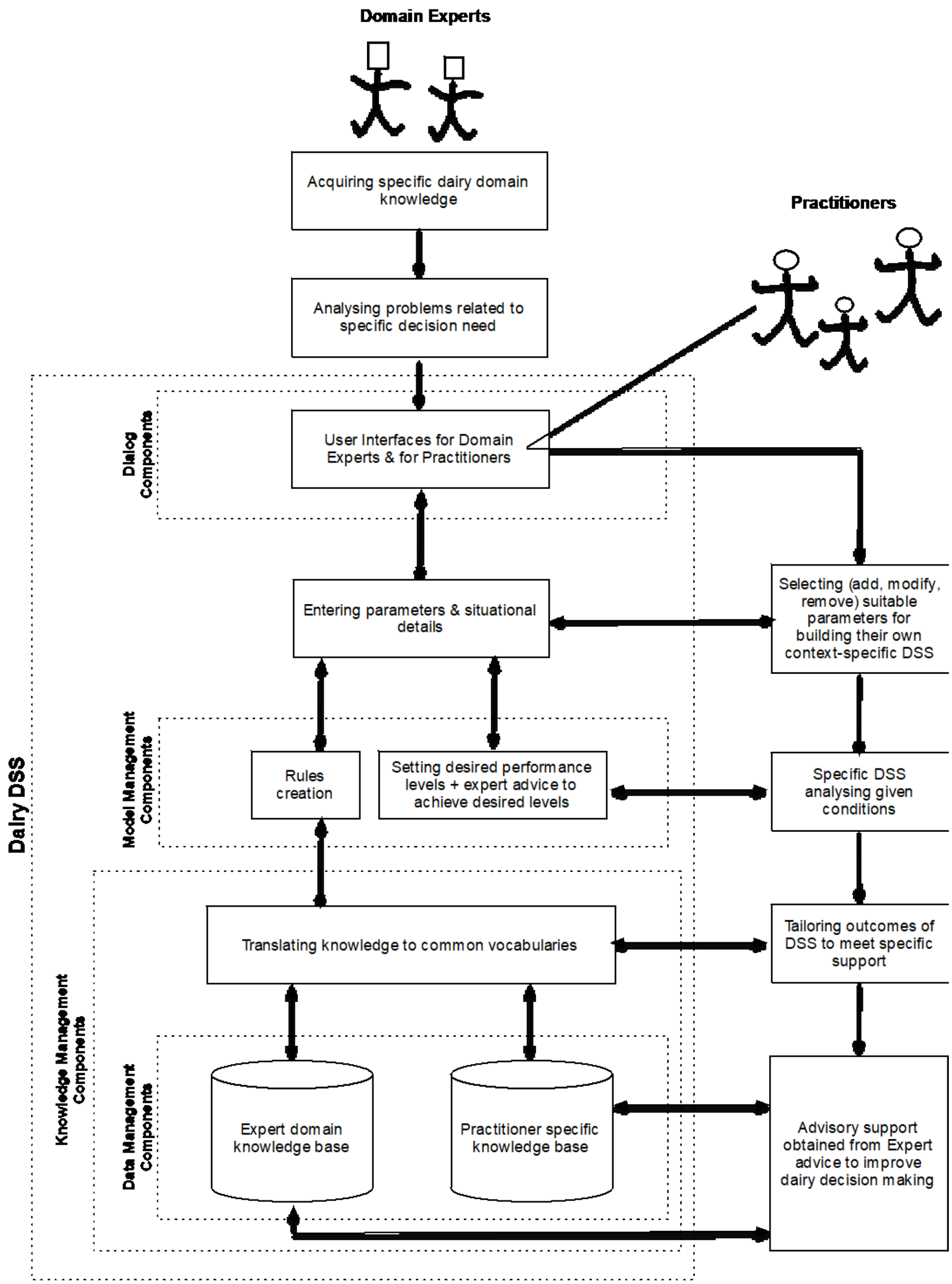


Figure 3: The Client-Specific Tailorable Dairy DSS

4.1.3 Reflection and Learning

Throughout the design process, feedback from participants was noted, was jointly reflected upon in facilitated focus groups and individual meetings, and a clearer understanding of the concerns and (meta) requirements emerged. This interaction articulated requirements for the Dairy DSS, but we were also mindful of the emerging metadesign and design principles.

First, the decision problem was semistructured, in that some factors were well-known (scientific knowledge), while other aspects relied on human judgment and tacit knowledge or were not completely understood (e.g., the impacts of local contextual variability). Second, for certain aspects of the problem, scientific knowledge expressed in the form of rules could be identified as the basis for decision support. Third, decision support was required for decision-making by practitioners in dynamic contexts: practitioner knowledge was paramount in mediating scientific knowledge and rules and in catering to contextual variability. Finally, knowledge sharing and translation and joint activity between the DE(s), practitioner(s), and the DSS designer were necessary for designing decision support appropriate to practice. Figure 3 provides an overview of the final Dairy DSS artifact.

4.1.4 Formalization of Learning

For this initial tailorable design to be applicable to other DSS contexts, it was important to identify the metarequirements to which this design solution might apply. Reflection on salient aspects of the “dairy” problem led to defining the metarequirements as follows:

- Semistructured primary production decision problem
- Supporting capture and representation of scientific knowledge and practitioner knowledge
- Requiring knowledge sharing and translation among the DE(s), practitioner(s), and the DSS designer
- Needing tailorability by practitioners because of dynamic contextual conditions
- Catering to dynamic, context-specific requirements of practitioners.

Characterizing the problem class for which the implemented DSS was a specific solution enables reflection and debate on design principles critical to designing practitioner-centric DSS (including user requirements, principles for selecting system features, and principles deemed effective for guiding the development process (Walls et al., 1992). Considering the metarequirements, working both from the emerging ideas in the spreadsheet (Figure 2) and from the DSS model (Figure 3), we articulated design principles and a metadesign, arguably suited to application to any specific instance of this problem class. The meta-artifact is detailed in Figure 4 showing five tentative *design principles* (DPs).

The set of emerging design principles (DPs) from Cycle 1 are detailed below.

DP1: Design to support collaboration enabling knowledge sharing and joint action among domain experts and practitioners. Prior to the start of secondary design activities by practitioners, collaboration for knowledge sharing between domain experts and practitioners is essential to populate the knowledge base with established knowledge components, including the parameters of decision support. This frames the decision space based on criteria from both current science and practice. As new science is discovered, specific parameters may become more or less relevant and new ones may be introduced by domain experts. Similarly, practitioners may or may not have success with particular settings or local feed combinations, which is quantifiable in terms of yield or other production measures. Thus, in turn, data relevant to learning from the practical outcomes of decisions feedback into the knowledge base.

DP2: Design functionality that combines scientific knowledge and practice-based knowledge for creating and re-creating decision support rules. Rule creation and re-creation are essential activities of the domain expert in the DSS design environment. DEs are responsible for initially entering parameters in order to develop the domain knowledge base by identifying concepts, factors, and their relationships defined through the ontology that provide generic structure for the decision-making scenario. System features are required so that predefined concepts, sets of terms, and factors and relations from the knowledge base can underpin decision-making rules for the DSS. The applicability of these rules remains a practical choice for the end user—any rule-based system implies a potentially infinite regress of metarules addressing all anticipated conditions: an unrealistic design aim. While scientific knowledge provides relevant input, the practitioner must qualify, weigh, and assess its utility in the active application context. Equally, feeding back lessons from practice updates scientific understanding.

DP3: Design for knowledge harmonization using familiar vocabulary. Different knowledge bases require conceptual harmonization (i.e., equivalencing scientific knowledge or jargon entered by the domain experts with practitioner vocabulary-in-use for a particular domain). For effective knowledge representation, the domain ontology technique was used so that specialized terms could be translated or added to a common domain-specific vocabulary. Ontology approaches also provide options for describing tasks or domain knowledge components to facilitate reusability and sharing of such knowledge. The DSS designer and DEs add established technical terms when the knowledge base is initially produced through the knowledge acquisition process. Similarly, if practitioners later add new, local, concepts with a relationship to factors for DEs, they must provide details or definitions of those concepts. While a certain amount of “second language learning” may be implicated on both sides, it is essential that the understanding of terms is common. The shared nature of conceptualization in an ontology provides for the development of a common vocabulary, so that both user groups have mutual or equivalent terms for knowledge sharing. A co-design process involving representative stakeholders throughout development allows such terms to be clarified, agreed upon, and published.

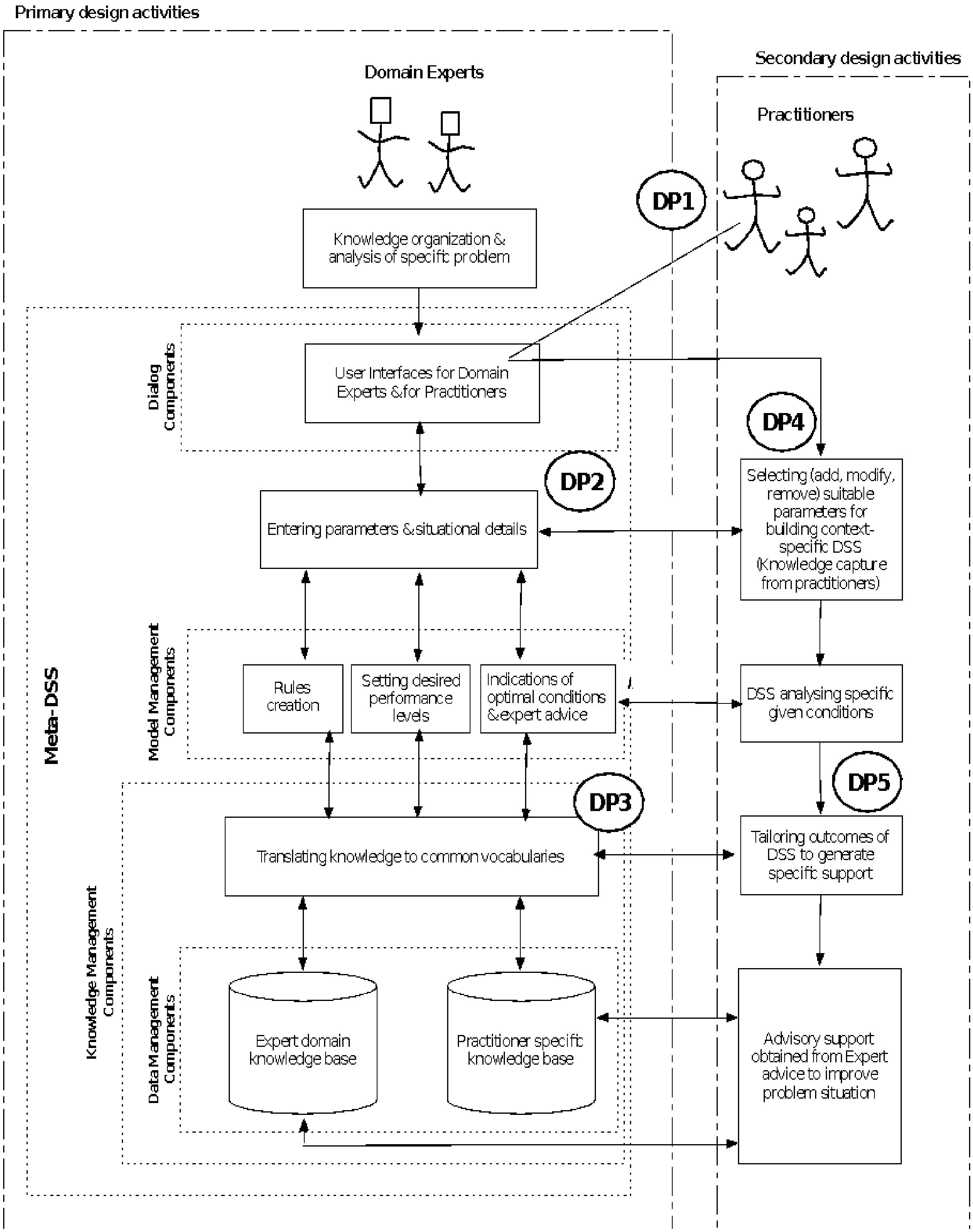


Figure 4: Meta-Design (Practitioner-Centric DSS Design Environment: Version 1)

DP4: Design customization features for variable dynamic conditions within the problem context. Allowing practitioners control over some components is vital to allow adjustments for variable specific local conditions. Customizations are specifically expected and tinkering (Germonprez et al., 2011) is designed for, but other components, such as scientific or other constraints, are not modifiable by end users. Therefore, relevant system components are designed to be adjusted by the practitioner according to their context, choosing from options outlining possible courses of action or adjustments to farming practices, and advising the costs and benefits associated with each of those options.

DP5. Design procedural components to ensure secondary designer interaction with relevant knowledge. The functional process of secondary design ensures practitioner interactions with all relevant knowledge areas for building their own specific decision support applications. The process guides practitioners through each step, such as selecting an appropriate set of parameters suitable to their own context. Parameters can be added, modified, or removed to better capture the context of decision support problems. When the practitioner adds new parameters, the details of these changes go to the domain experts to ensure appropriate knowledge acquisition. Some restrictions are required for practitioners such that they cannot change rules or relationships between parameters and relevant factors unless permission is granted or a meaningful relationship is validated by domain experts. The .NET implementation, and online/cloud platforms in general allows for monitoring, knowledge updates, data collection. and ongoing communication among stakeholders as designs evolve or emerge, thus overcoming the static limitation of predesigned systems.

4.2 ADR Cycle 2: DSS for Foresters

While we have proposed some design principles and theory, we need a complementary study to validate these further in another domain. In the second cycle we sought to test and validate the design environment's metarequirements, metadesign and design principles from Cycle 1 in a different instantiation context—Strategy 1). The comments made in Table 3 suggested that the artifact had mutability for customizations of design for emerging specific contexts and the features had potential applicability to other primary production sectors. In Cycle 2 we again essentially followed the ADR stages of iterative building and evaluating, so this will be detailed only briefly, highlighting slight differences in approach (Figure 1).

3.2.1 Problem Formulation and Matching

The Department of Employment, Economic Development and Innovation (DEEDI) has responsibility for developing and promoting the forestry industry in Queensland, advising and supporting foresters in the field. One particular concern has been providing scientific advice relating to pest infestation management and control. A practice had developed of foresters sending photos of pests via mobile devices to scientists, asking for identification and advice: their preference was to receive almost instant feedback so they could immediately start applying appropriate treatments and controls. This placed considerable burden on the scientists, who decided to work with a DSS designer to build a DSS for foresters that encapsulated their scientific knowledge. This system had proved to be cumbersome and was not greatly utilized by foresters. The foresters had been very critical of the DSS, arguing that the knowledge base was inadequate, there was little flexibility in the system, and there was no ability to tailor the system according to local changeable conditions. After attending a presentation on the Dairy DSS project, DEEDI approached us for advice and help in building a new DSS.

Initial investigations suggested the decision context was analogous to that in Cycle 1 and provided a complementary domain for a similar problem class (see Table 4). The system was intended to support foresters' (practitioners) decision-making about pest infestations and suitable controls. Scientific knowledge of the forestry scientists (domain experts) in terms of pest identification, treatments, treatment effectiveness, and advice for practitioners was elicited and stored in both text and graphic forms. As pest management knowledge is continually evolving, the DSS needed to periodically update the knowledge base effectively to support practice. This scientific knowledge was melded with practitioner knowledge relating to climatic and terrain influences, the efficacy of various treatments aimed at minimizing tree, leaf, and timber damage in specific locations, new types of insects and damage emerging, and the like. The decision-making context involved dynamic influences (for example, climate, seasonality, rainfall, tree age, maturity, type, space between trees); therefore, configurability of the DSS by the practitioners was, again, an important design concern. In addition to working with clients to ameliorate their perceived problems, our research interest focused on considering whether the meta-artifact and design principles from Cycle 1 were appropriate to support the development of a DSS for foresters. From a design research perspective, this Forest DSS was important for evaluation of the metadesign and design principles, and also to establish whether design knowledge developed in one context could be utilized in another.

Table 4: The Design Cases' Similarities and Differences

Similarities	Differences
Semistructured, decision support required • Both Dairy and Forest DSS involved semistructured decision problems • Scientific knowledge of the domain experts (agricultural scientists and forestry scientists) is well defined • Other aspects of the decision context rely on judgment, tacit knowledge, or were not completely understood (the impacts of local contextual variability)	**Context** • Dairy DSS: Dairy industry, decisions affecting milk protein concentration • Forest DSS: Forestry industry, decisions about pest management and control
Rely on capture and representation of scientific knowledge and practitioner knowledge *In both Dairy and Forest DSS:* • Scientific knowledge to enhance operational practices and support is required • Practitioner knowledge is included alongside scientific knowledge.	**Project background** • Dairy DSS: Improve farmers' decision support, particularly to maximize profit at minimal cost, enhance milk protein concentration, thus enhance productivity • Forest DSS: Minimize potential for losses associated with pest infestations in plantations; alleviate operational staff issues, such as lack of monitoring time due to heavy workloads and high turnover rates necessitating frequent training, underreporting of pest infestations, and failure to record pest-free areas
Knowledge sharing and translation *Both Dairy and Forest DSS involved* • Sharing of knowledge between scientists and practitioners • Knowledge transfer and translation capabilities were required to facilitate this, especially to ensure understanding by practitioners • Domain experts to input translation so that it was understandable by practitioners required	**Project motivations** • Dairy DSS: Farmers struggling with economic and market changes associated with industry deregulation. A DSS solution can assist to explore cost reduction while increasing milk production. • Forest DSS: Rapidly evolving knowledge of pest management in forestry, problems of training of field staff, DSS tool can provide comprehensive advice on the most appropriate pest management approaches
Tailorability by practitioner *Both Dairy and Forest DSS exhibited:* • Local variability and specific contextual conditions requiring practitioners to be able to tailor DSS to meet their specific subjective requirements	**Practice outcomes** • Dairy DSS: Administrative and field operation • Forest DSS: Field operation only

Thus, in addition to formulating the issues facing the DEs and practitioners with a view toward designing a DSS, there was a process of matching metarequirements, and metadesign from Cycle 1 against the evolving problem formulation. We summarize key similarities and differences in Table 4.

3.2.2 Instantiate, Evaluate, and Modify (IEM)

During the second stage, there was again an iterative approach to instantiation of the meta-artifact, formative evaluation by DEs and practitioners, discussion of the feedback, and modification of the DSS. The crucial difference to Cycle 1 came about because we were utilizing the meta-artifact (Figure 4) evolved from the Dairy DSS, populating it with relevant knowledge, parameters, and decision rules from the DEs (forest scientists) and the practitioners (foresters). Similar emphasis was placed on managing the group processes effectively as in Cycle 1. This approach follows the action research process, but also incorporates consideration of instantiation validity (Lukyanenko et al., 2014) to show the relationship between artifact features and the proposed design principles. We used focus groups to evaluate the reusability of the design artifact and to elicit scientific knowledge of the DEs, and how that knowledge might be utilized in the field by practitioners. DEs determined the factors, parameters, and result variables—together with the rules development—that allow result variables to be calculated from the factors and parameters. These were then entered into the structure of the metadesign. Working closely with practitioners enabled identification of a range of characteristics of a forestry site, such as the spacing between trees, pest species identified, and proportion of foliage damaged, all needing to be configurable in the DSS by practitioners according to dynamic local conditions.

Table 5: Feedback from Forest DSS Participants

Key aspects	Subcriteria	Comments
Design product	Final DSS	"We're deliberately tailoring the DSS by adding parameters and other settings, so that we can get some answers to our issues in the field" (P-f1). "The system supports quantitative assessment and 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 for addressing their need in the field" (P-f2). "The prototype is honed to the problem that we want" (P-f2).
	Strengths	"So it very much suits where we want to go, as far as what our priorities are, for the DSS" (DE-f1). "We can fill in the DSS with parameters and their details…it's also good in identifying what the gaps are, by providing the current details [field data], and where we need to go [options for improving productivity]" (DE-f2). "The system can be deliberately tailored to make it situation specific and remove the onus on users to determine what causes damage; instead, they can focus on the extent and type of damage for further treatments" (DE-f1).
	Improvements needed	"From a forestry perspective … an innovation such as being mobile would be really important for us…Anything to make the field guys' work easier" (P-f2). "It would be useful if there [were an] option for predicting about the future impacts of pest damages…" (DE-f2). "There's a whole range of decision-making issues and problems and things in forest health, and it's just a question of which one we choose to then go down the path with a lot of analysis for developing a complete knowledge base of the system...it would be good if the system integrates to a simulation model—that is very important when we are trying to predict population dynamics" (DE-f1).
	Value	"The system can store enough data for, say, the leaf beetle up here...if you're say looking at the population dynamics of a beetle and we've got the resources [pdf file and expert advice] provided by the system, it is really useful...more resources mean higher quality to us" (P-f1). "After looking at the use of the system, I think the system's flexibility will allow us to start with the data we have, and for further knowledge acquisition, and we know that we've got the industry support to do that" (DE-f2).
	Utility	"We are deliberately tailoring this to make it more simple and based on what a forester requires for their operation…to identify what causes the damage—they [foresters] just need to record the type and extent of the damage" (DE-f2). "I think the system will be handy for field operation and will have use for us to test many practical issues…industry profit is the ultimate aim of knowledge transfer" (DE-f1).
Design process		"A good thing about the process is it has the generic capacity to hold field situation details and it is also easy to modify all of them if changes are there due to seasonality" (DE-f1). "I think the whole approach fits pretty well with our current state of knowledge, and the main benefit of the process is that it gives a way of transferring knowledge electronically to the foresters and they can also add parameters to show us their need" (DE-f2). "The generic process is good enough for storing and generating data from the knowledge base, which I would suspect is maybe the one that industry would be most keen on to start with" (DE-f1).
Note: P = Practitioners; DE = Domain experts, f (1 to N) = Foresters		

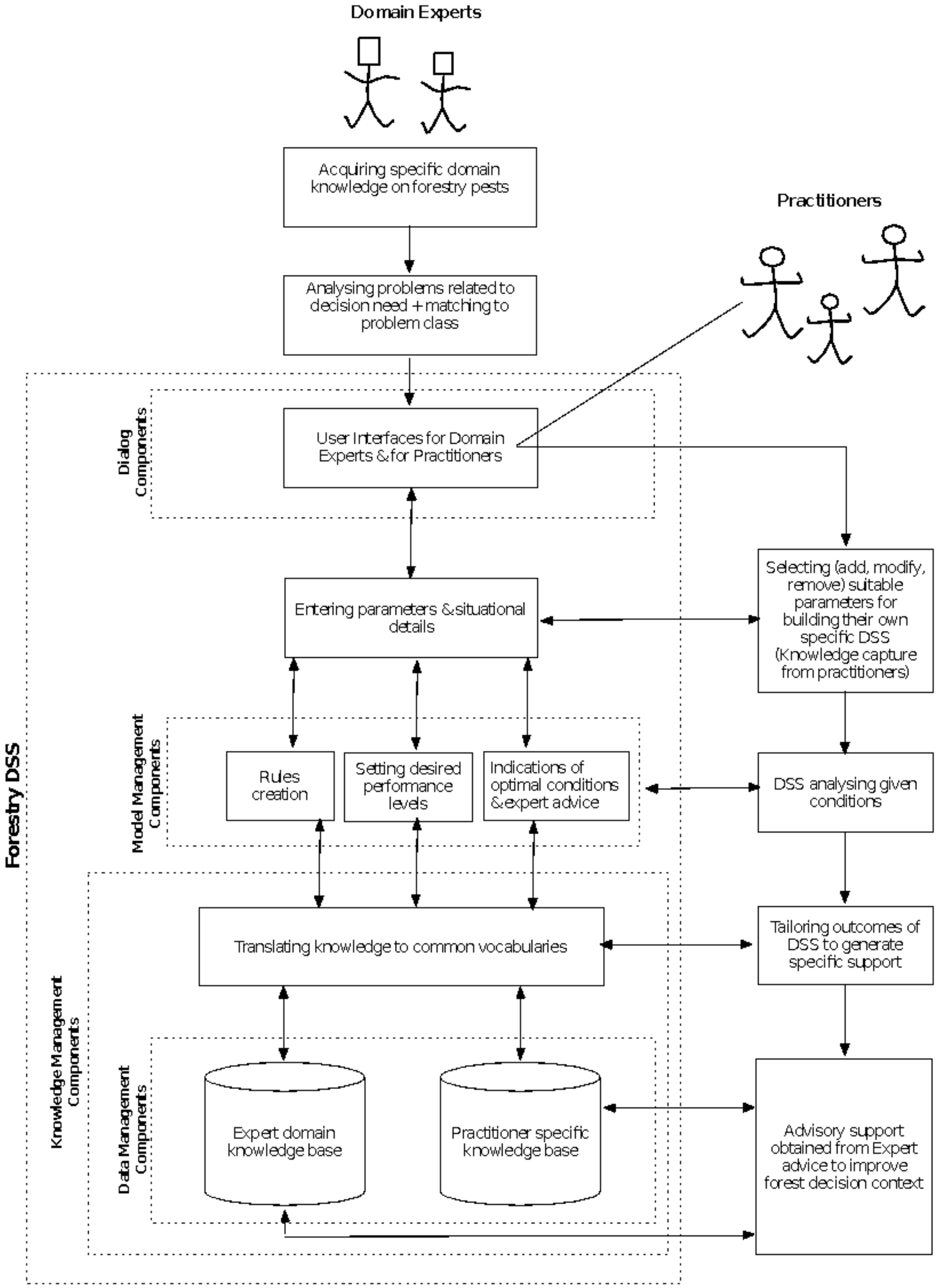


Figure 5: Forestry Pest DSS Artifact

The DSS provided some recommended actions to address the identified problem, and, where possible, allowed practitioners to access PDF files containing photographs of infestation symptoms and descriptions of insecticide treatments.

A prototype based on the metadesign was quickly built, and both DEs and practitioners were encouraged to experiment with this system and provide feedback, which was iteratively incorporated into the prototype. At the final focus group meeting, we also used a short feedback survey with participants to elicit their thoughts about the DSS developed and the process of their contribution to the design. We also conducted interviews with the participants. The evaluation of the

DSS was once again positive (see Table 5), indicating that the DSS design environment (metadesign) had been successfully instantiated in a different context.

4.2.1 Reflection and Learning

Matching of the new problem context to the metarequirements demonstrated that the forestry problem was very similar in type to the dairy problem. The problem domain was again deemed to be semistructured, with some well-defined applicable scientific knowledge but also some contextual uncertainties. Various aspects relied on human judgment, experience, and tacit knowledge, or were not completely understood (identification of pests and the impacts of local contextual variability, for example). In addition, some parameters changed rapidly and hence required regular adjustment.

However, we noted some differences in the forestry context. The decision-making problems in the problem domain proved to be much more complex than the dairy domain for a variety of reasons. First, the decision-making parameters were relatively uncontrolled and based on interconnected environmental factors. For example, the dairy farmers have a great deal of control over the amount of food, water, and shade provided to their cows: by contrast, the foresters have almost no control whatsoever over pests that might land in their forest, the speed with which they spread, the damage they cause, and so on. The decision support process for the foresters is not limited to providing expert advice in terms of identifying pest and disease symptoms and relevant control mechanisms. Foresters require comprehensive decision support in the field and that involved "pushing" out more specific scientific knowledge (e.g., digital images and text files) captured from the DEs and multiple other sources. Second, unless the practitioner had a lot of experience, diagnosis of the problem could be challenging. For example, the same type of insect can appear to be a different color, depending on the stage of its life cycle, seasonality, or for camouflage purposes. Complexity caused by these sorts of variations (seasonal, regional, climatic, and life cycle) requires a lot more science-based knowledge at the early stage to diagnose and identify the problem so that the forester can define situation specific parameters and accurately enter inputs into the application. Because of this, the DEs needed to provide more indicative detail into the knowledge base, to support both diagnosis and remediation. Third, throughout the instantiation process in the case, we found that participants, in this case, required current states of pest infestation to be assessed for impact on future production, thus adding to the complexity. Because of the dynamic nature of the problem context, the capture and representation of scientific knowledge and practitioner knowledge assumed greater importance, as did the need for tailorability of the DSS by practitioners catering to the situational requirements.

We found that the metadesign could be applied to this new context. However, the Forest DSS relied even more on the capture of scientific knowledge and practitioner knowledge to help manage the greater complexity. The dynamic contextual conditions in the field underscored the need for tailorability of the DSS so that practitioners can cater to the situational requirements.

4.2.2 Formalization of Learning

Despite the increased complexity, the metadesign outlined from the Dairy DSS was entirely applicable in the Forestry DSS context for meeting the decision support requirements of practitioners. Minor technical improvements were required so the model management components could support the upload of PDF files, such as scientific fact sheets and digital photos, into the knowledge base for the practitioners, in particular. This enabled better decision support to be provided to practitioners in the field (Figure 6).

5 Metadesign Theory

In this section, we specify the metadesign framework and evolved principles for designing a tailorable technology. The framework embraces primary and secondary design. The primary state encompasses knowledge management components, data management components, model management components, and dialog management components. The first phase includes DSS technologies (Meta-DSS in Figure 6), supporting a primary design state ensuring that domain knowledge and agreed decision criteria are accommodated in the knowledge base using a rule-based technique. The second phase makes the component technologies available for secondary design so that practitioners can create and re-create their own applications, adjustable to specific decision target outcomes. The metadesign offers ongoing support in configuring options for relevant and dynamic business scenarios and allows for saving emergent secondary designs as unique artifacts, while the embracing architecture controls integrity.

We followed the model suggested by Walls et al. (1992) for building design theory using "action research, coupled with iterative hypothesis development" (Markus et al., 2002, p. 187). Based on this, we revised the five design principles (DPs) articulated in Cycle 1, reflecting learning from the two design cases and making them more applicable to other problem contexts that share analogous requirements. The revised principles are reflected in the generic metadesign framework (Figure 6) discussed below, and following Walls et al. (1992), we summarize the design theory in Table 6.

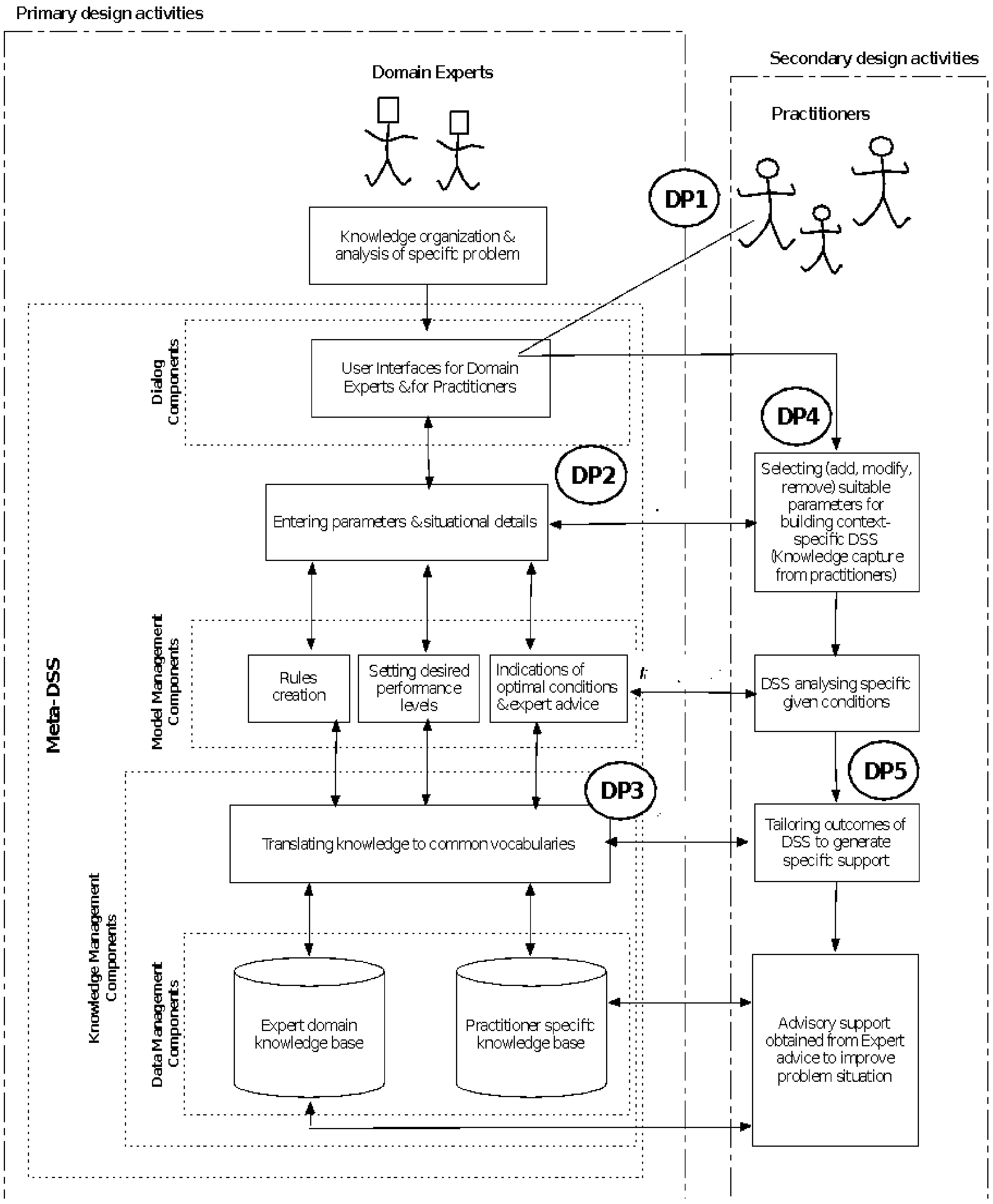


Figure 6: Final Metadesign: Practitioner-Centric DSS Design Environment

Table 6: Design theory for Tailorable DSS

Theory component	**Description**	**As evident in this research**
Design product		
Metarequirements	Class of problems to which theory applies	Domain has established structural aspects; settings of generally accepted parameters; relevant, mainly stable, scientific knowledge is available; practitioner-centric, tailorability, knowledge sharing and translation in decision support contexts
Metadesign	Class of artifacts that meet metarequirements	Personal, practitioner-centric, tailorable DSS that supports knowledge sharing and translation among domain experts and practitioners
Kernel theories	Theoretical grounding of the design product (Goldkuhl, 2004)	DSS theory: Keen (1980); secondary design: tailorable systems (Germonprez et al., 2007); common knowledge vocabulary: ontology theory (Studer et al., 1998)
Testable design product hypotheses	Are metarequirements satisfied?	Reflection and evaluation of artifact in use setting, of practitioner satisfaction and attitudes toward adoption, match between practitioner expectations and DSS artifact and generalization of solution instance.
Design process		
Design principles	Procedures adopted for design and co-constructing artifact.	1. Design for ongoing collaboration and action. 2. Design to support continuing knowledge acquisition and sharing. 3. Design for effective domain knowledge translation. 4. Design for context-specific secondary design. 5. Design to ensure integrity in emergent artifact design.
Kernel theories	Theoretical grounding of the design process	Tailorability (Germonprez et al., 2007); Communities of knowing (Boland & Tenkasi 1995); Collaboration across practices (Carlile, 2004)
Testable design process hypotheses	Does DSS design method produce artifact consistent with metadesign?	Reflection and evaluation of perceptions of artifact effectiveness and development process.

Table 7: Details of the Revised DPs

DP1: Design for ongoing collaboration and action	Co-design is needed to ensure that the initial knowledge base is formed using relevant and understood categories prior to secondary designing. It can evolve, as described previously. Mechanisms for continuing development as science and practice-based knowledge emerge are needed. These involve parameter input, revision, and testing, and continual monitoring of decision scenarios for relevance. Usage, access, and secondary activity can all be monitored in an online implementation and analytics can be used to inform future developments.
DP2: Design to support continuing knowledge acquisition and sharing	Appropriate knowledge acquisition functionality is important for domain experts. Features are needed to continuously capture domain knowledge for evolving the knowledge base so that practitioners can obtain current knowledge in terms they can procedurally use as components in secondary design. Such features provide an effective way of transferring the latest scientific research, productivity-oriented knowledge, and standard market and regulatory knowledge to the practitioners. Likewise, with the secondary design activities, knowledge sharing is important for practitioners to input specific farming knowledge for further utilization—especially to create or modify decision support rules. When the practitioners enter new parameters or specific settings through secondary design activity, a domain expert (DE) can ensure that appropriate relationships are drawn and identify the impact these new parameters have on outcome production variables (through the model management components). In consultation, DEs develop the domain knowledge base initially by identifying concepts, factors, and their relationships. Practitioners' inputs (through co-design and ongoing knowledge-sharing features) also contribute to validations in context. The predefined concepts, set of terms, factors, and relationships in the domain knowledge base are used to scope decision-making rules for the practitioner's decision support application, centered on taking the practitioner perspective, since this is where decisions are made and enacted. This ensures that relevant secondary design artifacts can continue to emerge as the context changes. Through subsequent exchange, the DE can effectively assist practitioners directly in building their specific DSSs, while increasing user engagement and empowerment in the DSS design process.
DP3: Design for effective domain knowledge translation	DSS designers should embed system features for knowledge translation necessary to reduce potential misunderstandings between designers, domain experts, and practitioners. In addition to clarifying definitions or intended interpretations of critical vocabulary, different knowledge types may require conceptual harmonization (i.e., equivalencing or synchronizing domain knowledge and jargon with practitioner vocabulary in use). The shared nature of conceptualization in an ontology provides for development of a common vocabulary so both user groups have mutual or equivalent terms for knowledge sharing, entailing perhaps an element of "second language learning". Ontology approaches facilitate common vocabularies and provide options for describing tasks and domain knowledge components to promote reusability and sharing. Scientific knowledge entered by the DE's translation to a common terminology facilitates the practitioner's interpretation of everyday operations (e.g., farming), much as a doctor might explain a medical condition to a layperson, and vice versa. Methodologically, co-design involving representative stakeholders is useful for establishing agreements on terminology and meanings within a defined context and can surface tacit assumptions and implicit models of a problem.
DP4: Design for context-specific secondary design	The practitioner's context-specific knowledge must be accommodated in the knowledge base for two distinct purposes. First, during the primary design activities, when the DSS designer and DEs initially prepare the knowledge base, practitioners can provide details of practical contingencies and relevant parameter range values so the DE can enter these as categories or knowledge components that will inform decisions in situated contexts. Second, apart from customizing parameter settings and entering values based on local specifics, (tinkering) the DSS design environment includes features for adding, removing and modifying parameters by the practitioners, (tailoring) which is verified for relevance and impact on outcome variables. Adjusting settings allows "what-if" options to be explored within the predefined problem space while tailoring more radically redesigns of the decision basis to specific practice, and systems may emerge that introduce new or omit prior relevant determinants. The artifacts are thus emergent, and mutable over time as they more closely fit changing contexts and practices. This principle both respects practitioners' knowledge and empowers them as designers, and avoids imposing a fully preconceived design, possibly fated to be ignored as irrelevant.
DP5: Design to ensure integrity in emergent artifact design	The DSS design environment provides secondary design functionality for practitioners to tailor parameters and build their own specific DSS application. During any tailoring activity, the primary design components of the DSS design environment remain in their original state, as the practitioner designs the secondary state. Practitioners changing parameters in a decision model however suggests monitoring and validation by domain experts, and in cases where outcomes prove practically successful, they may be treated as "hypotheses" for investigation and ongoing model

	refinement in another development cycle. Tailoring does not extend to violating definitional or other constraints, however, just as choosing wallpaper does not affect a room's structural integrity, which is ensured by overall architecture. The ontology developed in co-design of the original knowledge base is maintained throughout, and if new terms are added by either the DE or a practitioner, these would be mutually reviewed before extending or otherwise revising the ontology. Similarly, with rules representing the knowledge: facts and rules established from science or regulatory compliance would not be alterable in secondary design: new rules (e.g., from legislation or market information) would be incorporated in the primary design modules, and communication initiated from the DE side. Equally, practitioners may suggest a new rule or influence on an outcome variable—this can also be incorporated following mutual review and validation.

The diagram and tables comprise the proposed theory, and perhaps little further descriptive narrative is necessary, other than the consideration of the theory's evaluation and generality. Pries-Heje et al. (2008) distinguish ex post (after artifact construction) and ex ante (before construction) evaluation strategies and also distinguish naturalistic evaluation (real users, real systems, real problems) from artificial evaluation, which is usually intended to test utility or design hypotheses. These concepts were variously involved in the approach taken here. While the emerging theory's utility firmed up as the work progressed, any theory is provisional, and full evaluation of a theoretical artifact inevitably involves an extended history of usage with adaptation as required. As a first pass, however, the criteria suggested by Iivari (2015) can be applied immediately. For Strategy 1 aspects, successful instantiation provides a proof of concept, for which Hevner et al.'s (2004) guidelines offer five classes of possible summative evaluation techniques: observational, analytical, experimental, testing, and descriptive (which are used and more fully described in the original studies). Our focus groups made comments (Tables 3 and 5) concerning the potential for adoption, which Purao and Storey (2008) describe as an essential addition to Hevner et al.'s list. This is in addition to the formative feedback gained during construction iterations described earlier—which, though more artificial than naturalistic, incorporated representative users "playing" with the system in realistic scenarios. Since both rigor (correctness of output, replicable development methods) and relevance (focus groups of stakeholders, ongoing attention to industry context during development) were applied, this suggests that the case study components informing the theory proposed are inherently robust, with utility and applicability.

The artifact being evaluated here, however, is a theoretical one, representing a "conceptual IT meta-artifact (as a DSR contribution)" (Iivari 2015, p. 109). In considering appropriate evaluation Iivari (2015) questions whether testing in the same client context is justified, and notes that in theory testing it is better practice to use a different data set than that used for theory building. The tentative design principles emerging from the first study were refined and essentially validated. Further theory testing would strengthen confidence, through convergence to a stable form. By showing that the initially successful development of design theory transfers from one client context to another, however, we provide a general solution concept suitable for such further evaluation and testing.

Generalizability is a challenge in DSR, especially when specific contexts and problem constraints inherently apply. Sein et al.'s (2011) approach is to reconceptualize both the problem and the solution instances as belonging to specific classes and then to articulate the learning as design principles as generalized outcomes connected to these classes. We followed this approach throughout this research, and comment further in the discussion below. Iivari (2015) specifies a very similar approach for generalization under Strategy 2, where specific problems encountered during a real development are generalized into a class of problems, and lessons from developing a specific solution are generalized into a meta-artifact as a general solution concept (such as design principles), which is then associated with the class of problems identified.

6 Discussion and Conclusion

Using design science across two different client contexts we developed a DSS design environment, allowing specifically tailored DSS to be secondarily designed in the client contexts. We used secondary analysis to extend the cumulative learning into a more general design theory applicable to decision problems sharing analogous metarequirements and specifying five design principles.

The proposed metadesign represents a shift in the approach to DSS design and development, but one in the tradition envisaged by one of the field's founders. Our kernel theory is drawn from Keen (1980) who made the empirical observation that many uses of DSS could not be predicted when the system was designed, but also stated (p. 8) that the "Gorry-Morton framework is not a complete or convincing theoretical statement", partly due to the problematic construct of structuredness (see also Arnott et al. 2017). While we retain the terms structured and semistructured here, due to their currency as prevalent and understood ideas in the literature, we agree that these are nonempirical, fuzzy descriptive categories of essentially only heuristic value in identifying potentially suitable domains, tasks, and problems. The importance of Keen's work is in reifying DSS as a distinct class of information systems, with an associated scope and agenda—which, though some details may be dated, remains a classic theoretical framework providing a foundation for DSS research and artifact construction. Of particular relevance is Keen's focus on adaptive design, use and evolution to overcome the Gorry-Scott Morton limitations and also his assertion that if a system function does not have a direct relation to a concept in the user's mind then it "really cannot be used" (Keen, 1980, p. 20).

The metadesign we have proposed provides DSS designers with a design environment for addressing known DSS shortcomings, relying on co-design processes to embed distinct types of knowledge and empower secondary design for varying contextual conditions. Evaluations of the instantiations of the metadesign indicate stakeholder satisfaction with relevance and generalizability, responding to concerns (e.g., Hosack et al., 2012) that DSS should produce appropriate client-centric outcomes. The design environment increases quality and relevance where situation-based knowledge is critical to decision-making and allows for ongoing emergent artifact development. The representative user evaluations tabulated for the respective cases indicate positive user acceptance—due, in part, to ease of understanding functional use—and perceptions of its helpful potential. This is arguably due, to some extent, to the parameter terms fitting the practitioners' own vocabularies and mental models of how various factors interact, consistent with Kayande et al.'s (2009) finding that DSS designed to align the DSS model with the user's model will receive better evaluations—this idea is also embodied in our design theory.

In addition to the contribution to DSS research, our studies demonstrate the practical value of Iivari's (2015) Strategy 1 and 2 concepts by producing artifact(s) with high practical relevance and proposing a design theory articulated as a metadesign with distilled design principles applicable to a certain type of problem. By instantiating and evaluating robustness, utility, and applicability in a new client context, as Iivari suggests, we go beyond previous studies which were limited to a single domain.

There are, however, recognized methodological issues associated with using secondary analysis across multiple cases, which merit consideration in this context, especially as the method grows in use. First, although time, cost, and often quality advantages accrue from using already collected data and archiving provides opportunities to test different models, if the primary data was collected for a different research question, data sets may be incommensurable and their reuse may not fit the new research context (Donnellan, Trzesniewski and Lucas, 2011). Hammersley (2012), however, argues that lack of fit and access to original research context does not necessarily always apply to reused data and considers the suggestion that if data are reflexively constructed by research, analysis has the same "epistemic status (as) primary research" (p. 108). As Arnott et al. (2017) argue, for interpretivists the use/reuse distinction does not apply, while for other qualitative researchers there is no need to have similar research questions, and when similarities exist in data collection, studied phenomena, units of analysis, and research leads, cases can be combined for secondary analysis, as was the case here. The critical considerations around secondary analysis do not undermine the present research, which did not involve combining quantitative data sets but used qualitative data in an analytic expansion of the researchers' own work. As such, it retained privileged insight into the data sources and representation, such that no contradiction in interpretation between the present analysis and the prior primary studies emerged.

This work contributes to DSR literature, in specifying a metadesign theory for tailorable technologies, particularly applicable to an identified class of semistructured primary production decision support problem requiring both field and scientific knowledge. Successful extension to problems with analogous meta requirements is considered plausible, given clear scope and effective parameterization. Theoretically, we have also demonstrated one way of deriving design theory (DSR knowledge) from design cases. Practically, we have shown how disparate types of knowledge can be represented to support tailorability in practitioner context and instantiated artifacts for industry end users. In summary, our findings contribute to the complex artifact design knowledge through the tailorable provision of a design environment artifact for meeting practitioner's contextual demands. We also explored this understanding of artifact design for reuse potential in the production of useful DSS artifacts.

Our original research was an example of an IT-dominant approach (Sein et al., 2011) to DSS design that emphasized innovative technological design and there are some limitations within our study. While group processes and collaboration were important and were informed by theory (see Table 6), they were not originally highlighted. As the processes in knowledge transfer, translation, and representation are critical, we suggest that emphasizing perspective taking and relational competence across language games should be given more importance in development. Rules are not always an appropriate representation of knowledge, and so applicability may be limited to semistructured domains with relatively well specified scientific components. While early expert systems were brittle, working best in well-structured domains, they later became more usefully productive in scoped areas, as later approaches allowed for more adaptation—whether through introducing learning or discovering knowledge in large data sets, putting "humans in the loop" to adjust for context, or using case-based reasoning (CBR) rather than a rule base. Although recognizing that some domains are more suited to rule-based representations than others, we did not specify this as a constraining principle and believe other underlying knowledge models are also applicable. This claim, however, would also benefit from further examination.

Regarding generalizability to other domains, we contend that, conservatively, the theory applies to the design of (for example) beef cattle, sheep, or arable farming, but that it also plausibly applies without change to other domains (commercial, scientific, financial) characterized by structuredness, agreed-upon knowledge, parameterizability, and involved users. While Seins et al.'s (2011) ADR explicitly provides for the generalization of problems and solutions, defining what constitutes a class to which generalization can be made remains a challenge. CBR has an extensive literature on similarity measures used to classify one problem as being similar to another (e.g., Liao et al., 1998). While it is beyond the scope of this paper to discuss these, a future taxonomy characterizing decision problems might provide guidance beyond the heuristics of expertise. It is likely, however, that as structuredness in problem specification decreases, similarity criteria will rely more on heuristics and experience than on more formal mathematical mappings of equivalence. In our research, the design environment was conceptualized from the outset to be domain neutral in order to accommodate secondary designs across a range of decision contexts. The dairy focus group participants saw the potential for beef cattle farming decision-making, and when the forestry domain problem was presented by industry, the design team intuitively assessed it as a potentially suitable application to test and further formalize the emerging design theory. Further research, however, is needed to provide more detailed theoretical guidance.

Within primary production, future research could explore simulation and predictive analytic technologies within the primary model for predicting field decision outcomes as larger data sets become available. The metadesign itself is domain-neutral and tailorable, and extending it to other domains would be interesting to establish scope and generality. In addition, although implemented in a .NET environment, it is likely that future decision support will be furnished via app access to a cloud platform, allowing continuing updates (technical and knowledge-based), centralized monitoring, and analysis of practice, while removing from the practitioner the burden of maintaining specialized hardware and proprietary software. This would go toward organization-dominant (Sein et al., 2011) research design and aspects around this may also be profitably explored, with, for instance, implementation, risk analysis, and postimplementation expert analysis of data on decisions and outcomes that might further extend relevant domain and organizational knowledge.

Although Keen (1980) recanted the subtitle of "Decision support systems: An organizational perspective" (Keen & Scott Morton, 1978) he recognized "The difficulty of institutionalizing a system and embedding it in its organizational context so that it will stay alive when the designer/consultant leaves the scene" (Keen, 1981, p. 26). Further research is thus signaled that could clarify secondary designing and examine whether other technical or organizational elements are required in the design environment to improve secondary design activity. Relatedly, the role of the primary designer should be contextualized: while an academic or business analyst can produce a working prototype, implementation in organizational practice may require more substantial professional software engineering, including testing and integration with other systems as required. If so, a technical representative from the organization could usefully be involved in the design process. While we were fortunate to have access to domain scientists during development, we paid close attention to effective knowledge sharing through participatory development and did not rely on a nonexpert's knowledge elicitation process alone in domain modeling.

Although each DSS was demonstrated as a working prototype with all functions, these were developed only to a level consistent with stakeholder evaluation. The summative evaluation of the design products and process were conducted at a single point in time, immediately after the delivery of the final design, although formative evaluations occurred throughout. As the source studies applied ADR methodology serendipitously, ADR's desirable aspect of ongoing organizational validation in practice is weak, and a potential intention to use is not the same as authentic use. Future evaluation could be longitudinal, studying patterns of use and perceptions of value or utility over time, in naturalistic settings as well. This would increase confidence that ongoing relevance has successfully been addressed in this research.

In design science, a future research direction is also implied since the evolved design principles have not been further tested, which is beyond the scope of this paper. It is possible that the "final" principles could be formulated otherwise (fewer, more, or more generic in nature) and the class of problems to which the design framework applies could be more specifically detailed. While system evaluators saw direct application of the approach to analogous farming applications, these would need to be empirically validated in further research. While the design parameterization structure allows new artifacts to be tailored and adapted relatively quickly, the instantiation validity would be increased by more examples in such domains. By using an architecture that delegates much of the relevance aspect to secondary design activity, both development costs and feature complexity are reduced, potentially allowing multiple artifact instantiations to be experimentally compared, and further validating the links between theoretical principles and specific designs.

As well as technological developments, Lee and Baskerville (2003, p. 241) note:

> *The process of action research suggests that the ability of a theory to be generalized to a new setting could also depend on factors outside the theory itself [and…there is a social process for testing, refining, and hence circumspectly generalizing the theory to a setting where it was not previously developed or tested.*

Design science theory is still evolving, particularly for DSS design, and we hope that our work will encourage future researchers to progress both rigorous and relevant research in this field.

Acknowledgments

The authors would like to thank the Australian Research Council (ARC) for funding the initial DSS design studies and the Queensland Department of Primary Industries and Fisheries (QDPI) for their financial and in-kind support. The authors also gratefully acknowledge the senior editor and reviewers for providing valuable feedback to improve the paper.

References

Arnott D. (2006). Cognitive biases and decision support systems development: a design science approach. *Information Systems Journal, 16*(1), 55-78.

Arnott, D., & Pervan, G. (2008). Eight key issues for the decision support systems discipline. *Decision Support Systems*, *44*(3), 657-672.

Arnott, D., & Pervan, G. (2010). How relevant is fieldwork to DSS design science research? *Proceedings of the 15th IFIP WG 8.3 International Conference on Decision Support Systems: Bridging the Socio-technical Gap in Decision Support Systems: Challenges for the Next Decade* (pp. 108-119).

Arnott, D., & Pervan, G. (2012). Design science in decision support systems research: An assessment using the Hevner, March, Park, and Ram guidelines. *Journal of the Association for Information Systems, 13*(11), 923-949.

Arnott, D., & Pervan, G. (2014). A critical analysis of decision support systems research revisited: The rise of design science. *Journal of Information Technology, 29*(14), 269-293.

Arnott, D., Lizama, F., & Song, Y. (2017). Patterns of business intelligence systems use in organizations, *Decision Support Systems, 97,* 58-68

Bell, S, (1992). Information systems and developing countries. *Computer Bulletin, 4*(3), 24-25.

Boland, R. J., & Tenkasi R.V. (1995). Perspective making and perspective taking in communities of knowing, *Organization Science, 6*(4), 350-372.

Brézillon, P., & Pomerol, J. C. (1998). Using contextual information in decision making. *Proceedings of IFIP TC8 I WG8.3 International Conference on Context-Sensitive Decision Support Systems.*

Carberry, P. S., Hochman, R. L., Mccown, R. L., Dalglish, N. P., Foale, M. A., Poulton, P. L., Hargreaves, J. N. G., Hargreaves, D. M. G., Cawthray, S., Hillcoat, N., & Robertson, M. J. (2002). The FARMSCAPE approach to decision support: Farmers', advisers', researchers' monitoring, simulation, communication and performance evaluation. *Agricultural Systems, 74*(1), 141-177.

Carlile, P. R. (2004) Transferring, translating, and transforming: An integrative framework for managing knowledge across boundaries. *Organization Science*, *15*(5), pp. 555-568.

Carlsson, S. A. (2007). Developing knowledge through IS design science research: For whom, what type of knowledge, and how. *Scandinavian Journal of Information Systems, 9*(2), 75-86.

Chamberlain, A. (2006). *Protein Plus Checkbook: A guide to increasing milk protein % and profit.* Brisbane, Australia: The State of Queensland, Department of Primary Industries and Fisheries.

Cox, P. G. (1996). Some issues in the design of agricultural decision support systems, *Agricultural Systems, 52*(2), 355-381.

Churi, A. J., Mlozi, M. R. S., Mahoo, H., Tumbo, S. D., & Casmir, R. (2013). A decision support system for enhancing crop productivity of smallholder farmers in semi-arid agriculture. *International Journal of Information and Communication Technology Research*, *3*(8), 238-248.

da Costa Campos, F. D. B. V. (2015). The impact of user-generated content on Facebook advertising performance (PhD dissertation, Católica Lisbon School of Business & Economics, Lisbon, Portugal). Retrieved from https://repositorio.ucp.pt/bitstream/10400.14/20260/1/Dissertation_Francisco%20Campos_final.pdf.

Donnellan, M. B., Trzesniewski, K. H., & Lucas, R. E. (2011). Introduction. In K. H. Trzesniewski, M. B. Donnellan, & R. E. Lucas (Eds.), *Secondary Data Analysis: An Introduction for Psychologists.* Washington, DC: APA.

Edwards, A. (2012). The role of common knowledge in achieving collaboration across practices. *Learning, Culture and Social Interaction, 1*(1), 22-32.

Fischer, G. (1999). Domain-oriented design environments: Supporting individual and social creativity. In J. Gero & M. L. Maher (Eds.), *Computational Models of Creative Design IV* (pp. 83-111). Sydney, Australia: Key Centre of Design Computing and Cognition.

Friis, S. (1996). Information systems design: A user-involved perspective, In K. S. Gill (Ed.), *Human machine symbiosis: The foundations of human centered systems design*. Secaucus, NJ: Springer.

Gammack, J. G. (1999). Constructive design environments: Implementing end-users systems development, *Journal of End User Computing, 11,* 15-23.

Gammack, J. G., Fogarty, T. C., Battle, S.A., Ireson, N. S. & Cui, J. (1992). Human centred decision support: The IDIOMS Project. *AI and Society* *6*(4), 345–366.

Gasson, S., (2003). Human-centered vs. user-centered approaches to information system design, *Journal of Information Technology Theory and Application, 5*(2), 29-46

Germonprez M., Hovorka, D., & Collopy, F. (2007). A theory of tailorable technology design. *Journal of the Association for Information Systems, 8*(6), 351-367.

Germonprez, M., Hovorka, D., & Gal, U. (2011). Secondary design: A case of behavioral design science research. *Journal of the Association for Information Systems, 12*(10), 662-683.

Gillard, P., & Johnson, D. E. (2001). Fertiliser advisor dairy: An expert system for dairy pastures in Tasmania. *Proceedings of the Australian Agronomy Conference, Conference Design Pty Ltd.*

Gill, K. G. (1996). *Human machine symbiosis: The foundations of human centered systems design,* Secaucus, NJ: Springer.

Goldkuhl, G. (2004). Design theories in information systems-a need for multi-grounding. *Journal of Information Technology Theory and Application, 6*(2), 59-72.

Gregor, S., & Hevner, A. R. (2013). Positioning and Representing Design Science Research for Maximum Impact, *MIS Quarterly, 37*(2), 337-355

Gregor, S. & Iivari, J. (2006). Designing for mutability. *Information Systems Artifacts: Information Systems Foundations Workshop 2006*.

Hammersley M. (2012). Can we re-use qualitative data via secondary analysis? Notes on some terminological and substantive issues. In J. Goodwin (Ed.), *Secondary Data Analysis* (Vol. 3, pp, 107-118). London: SAGE.

Hayman, P. T., & Easdown, W. J. (2002). An ecology of a DSS: Reflections on managing wheat crops in the north-eastern Australian grains region with WHEATMAN. *Agricultural Systems, 74*(1), 57-77.

Hevner, A., March, S., Park, J., & Ram, S. (2004). Design science in information systems research, *MIS Quarterly, 28*(1), 75-105.

Holsapple, C. W. (2008). DSS architecture and types, In *Handbook on decision Support Systems 1: Basic Themes* (Burstein, F. & Holsapple, CW, (Eds.). Springer-Verlag, Berlin, 163-190.

Hosack, B., Hall, D., Paradice, D., & Courtney, J.F., (2012) A look toward the future: decision support systems research is alive and well. *Journal of the Association for Information Systems, 13*(5), 315-340.

Hovorka, D. S. (2010). Incommensurability and Multi-paradigm grounding in design science research: Implications for creating knowledge. In J. Pries-Heje, J. Venable, D. Bunker, N. L. Russo, & J. I. DeGross (Eds.), *Human Benefit through the Diffusion of Information Systems Design Science Research* (pp. 13-27). Berlin: Springer.

Iivari, J. (2015). Distinguishing and contrasting two strategies for design science research. *European Journal of Information Systems, 24*(1), 107-115.

Kane, G. C. & Fichman, R. G. (2009). The shoemaker's children: Using wikis for information systems teaching, research, and publication. *MIS Quarterly, 33*(1), 1-22.

Karmakar, S., Lague, C., Agnew, J., & Landry, H. (2007). Integrated decision support system (DSS) for manure management: a review and perspective. *Computers and Electronics in Agriculture,* 57(2), 190-201.

Kayande, U., De Bruyn, A., Lilien, G. L., Rangaswamy, A., & Van Bruggen, G. H. (2009). How incorporating feedback mechanisms in a DSS affects DSS evaluations. *Information Systems Research, 20*(4), 527-546.

Kersten, G. E., Mikolajuk, Z., & Yeh, A. G. (2002). Design support systems for sustainable development: A resource book of methods and applications. New York, NY: Kluwer Academic Publishers.

Kerr, D., Cowan, R. T., & Chaseling, J. (1999). DAIRYPRO: A knowledge based decision support system for strategic planning on sub-tropical dairy farms: I. system description, *Agricultural Systems, 59*(3), 245-255.

Kerr, D., & 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), 61-69.

Keen, P. G. W., & Scott Morton, M. S. (1978). *Decision support systems: An organizational perspective*. Reading, MA: Addison-Wesley.

Keen, P. G. W. (1980). *Decision support systems: A research perspective*, CISR No. 54, Sloan WP. 1117-11180. Cambridge: MA: Center for Information Systems Research, Massachusetts Institute of Technology.

Keen, P. G. W. (1981). Information systems and organizational change, *Communications of the ACM, 24*(1), Jan. 1981

Koehne B., Redmiles D., & Fischer G. (2011). Extending the metadesign theory: Engaging participants as active contributors in virtual worlds. In M. F. Costabile, Y. Dittrich, G. Fischer, A. Piccinno (Eds.) *End-user development: IS-EUD 2011*. Lecture Notes in Computer Science, vol 6654. Berlin: Springer.

Lambert, J., & Elix, J (2003). *Reshaping rural extension: New players—new roles*. Retrieved from http://www.regional.org.au/au/apen/2003/refereed/047lambertjelixj.htm.

Lee, A. S., & Baskerville, R. L. (2003). Generalizing generalizability in information systems research, *Information Systems Research, 14*(3), 221-243.

Liao, W., Zhang, T., Zhiming, & Claude, M. (1998). Similarity measures for retrieval in case-based reasoning systems, *Applied Artificial Intelligence*, *12*, 267-288.

Lukyanenko R., Evermann J., Parsons J. (2014). Instantiation validity in IS design research. In: M. C. Tremblay, D. VanderMeer, M. Rothenberger, A. Gupta, V. Yoon (Eds) *Advancing the Impact of Design Science: Moving from Theory to Practice*: *DESRIST 2014*. Lecture Notes in Computer Science, Vol 8463. Berlin: Springer.

Markus, M. L., Majchrzak, A., & Gasser, L. (2002). A design theory for systems that support emergent knowledge processes, *MIS Quarterly, 26*(3), 179-212.

Merot, A., Bergez, J. E., Capillon, A., & Wery, J. (2008). Analysing farming practices to develop a numerical, operational model of farmers' decision-making processes: An irrigated hay cropping system in France, *Agricultural Systems, 98*, 108-118

McCown, R. L. (2002). Changing systems for supporting farmers decisions: Problems, paradigms, and prospects. *Agricultural Systems, 74*(1), 179 -220.

McKay, J., Marshall, P., & Hirschheim, R. (2012). The design construct in information systems design science, *Journal of Information Technology*, 27, 125-139

Meensel, J. V., Lauwers, L., Kempen, I., Dessein, J., & Huylenbroeck, G. V. (2012). Effect of a participatory approach on the successful development of agricultural decision support systems: The case of Pigs2win. *Decision Support Systems 54*(1), 164-172.

Miah, S. J., Kerr, D. & von-Hellens, L. (2014). A collective artefact design of decision support systems: Design science research perspective, *Information Technology & People, 27*(3), 259-279.

Miah, S. J., Kerr, D., & 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,* 735-744.

Miah, S. J., Debuse, J., Kerr, D., & Debuse, V. (2010). A Practitioner-Oriented Decision Support Process for Forestry Pest Management. *Proceedings of the 21st Australasian Conference on Information Systems*.

Miah, S. J. (2009). End user as application developer for decision support. *Proceedings of the 15th Americas Conference on Information Systems*.

Miah, S. J. (2008). An ontology based design environment for rural decision support, *Unpublished PhD Thesis,* Griffith Business School, Griffith University, Brisbane, Australia.

Olesen, J. E., Trnka, M., Kersebaum, K. C., Skjelvag, A. O., Seguin, B., Peltonen-Sainio, P., Rossi, F., Kozyra, J., & Micale, F. (2011). Impacts and adaptation of European crop production systems to climate change, *European Journal of Agronomy, 34*, 96-112

Power, D. J., Sharda, R., & Burstein, F. (2015). *Decision support systems*. New York, NY: Wiley.

Power, D. J. (2012). Decision support systems: Concepts and resources for managers. Westport, CT: Greenwood.

Power, D. (2004). What is the difference between a DSS and a DSS generator? Retrieved from http://dssresources.com/faq/index.php?action=artikel&id=35.

Power, D. (2010). What are concerns about end-user DSS development? Retrieved from http://dssresources.com/faq/index.php?action=artikel&id=208.

Prasad, J. R., Prasad, R. S., & Kulkarni, U. V. (2008). A Decision support system for agriculture using natural language processing (ADSS), *Proceedings of the International MultiConference of Engineers and Computer Scientists.*

Pries-Heje, J., Baskerville, R., & Venable, J. R. (2008). Strategies for design science research evaluation. *Proceedings of the European Conference on Information Systems*.

Purao, S., & Storey, V. C. (2008). Evaluating the adoption potential of design science efforts: The case of APSARA, *Decision Support Systems*, *44*, 369-381.

Ren, Y., Kraut, R., & Kiesler, S. (2007). Applying common identity and bond theory to design of online communities, *Organizational Studies, 28*(3), 377-408.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). Why should I trust you? Explaining the predictions of Any Classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Ryder, G. (2009). The digital divide: Investigating the isle of man's dilemma using exploratory research to resolve the problem. *Proceedings of the European and Mediterranean Conference on Information Systems.*

Sein, M., Henfridsson, O., Purao, S., Rossi, M., & Lindgren, R. (2011). Action design research, *MIS Quarterly*, *35*(1), 37-56.

Sol, H. G. (1987). Paradoxes around DSS. *Proceedings of the NATO Advanced Study Institute on Decision Support Systems: Theory and Application*.

Sowunmi, A., Burstein, F. V., & Smith, H. G. (1996). Knowledge acquisition for an organizational memory system. *Proceedings of the 29th Hawaii International Conference on System Sciences*.

Studer, R., Benjamins, V. R., & Fensel, D. (1998). Knowledge engineering: Principles and methods. *Data & Knowledge Engineering*, *25*(1), 161-197.

Smirnov, A., Levashova, T., & Shilov, N. (2015). Patterns for context-based knowledge fusion in decision support systems. *Information Fusion 21*, 114-129.

Tarrant, A. (2016). Getting out of the swamp? Methodological reflections on using qualitative secondary analysis to develop research design, *International Journal of Social Research Methodology, 20*(6), 599-611.

Thorne, S. (1994). Secondary analysis in qualitative research: Issues and implications, In J. Morse (ed.), *Critical issues in qualitative research methods*, Thousand Oaks, CA: Sage.

Van De Ven, A., & Johnson, P. E. (2006). Knowledge for theory and practice. *Academy of Management Review, 31*(4), 802-821.

Valls-Donderis, P., Ray, D., Paece, A., Stewart, A., Lawrence, A., & Galiana, F. (2014). Participatory development of decision support systems: Which features of the process lead to improved uptake and better outcomes? *Scandinavian Journal of Forest Research, 29* (1),71-83.

Voinov, A., & Gaddis, E. J. B. (2008). Lessons for successful participatory watershed modeling: A perspective from modeling practitioners, *Ecological Modelling, 216*(2), 197-207.

Vizecky, K., & El-Gayar, O. (2011). Increasing research relevance in DSS: Looking forward by reflecting on 40 years of progress. *Proceedings of the 44th Hawaii International Conference on System Sciences.*

Walls, J. G., Widmeyer, G. R., & El Sawy, O. A. (1992). Building an information system design theory for vigilant EIS. *Information Systems Research, 3*(1), 36-59.

Walker, H. (2002). Decision support, learning and rural resource management. *Agricultural Systems, 73*(1), 113-127.

Winograd, T. (1995). From programming environments to environments for design. *Communications of the ACM*, 38 (6), 65-74.

Wittgenstein, L. (1953, 2001). *Philosophical investigations*. Oxford, UK: Blackwell.

Appendix A

Table A1. Hevner et al.'s (2004) Seven Guidelines as Used in the Dairy DSS Design Case

Hevner et al.'s DSR guidelines	**Dairy DSS artifact design**
Guideline 1: Design as an artifact Which form of artifacts to use? (e.g., construct, model, method, and instantiation).	An innovative artifact (software solution prototype) as instantiation was developed and field-tested
Guideline 2: Problem relevance Which particular problems are to be addressed by the innovative technological solution artifact?	Problem domain was identified that required the outlined software solution prototype. The problems addressed were based on business-critical decision-making in dairy operations.
Guideline 3: Design evaluation Which qualities are to be evaluated? (i.e., utility, efficacy).	A descriptive evaluation method was employed for prototype testing, both with industry users and other stakeholders, coupled with scenario analysis using secondary data.
Guideline 4: Research contributions To which body of knowledge does the design research contribute?	The models used for the decision outcomes within the artifact were developed by domain experts using practice-based knowledge. This knowledge has been used as a kernel to derive the decision outcomes by using constraint-based formulas.
Guideline 5: Research rigor Design must represent an application of rigorous methods in the construction and evaluation of the design artifact.	Rigor was achieved through expert scrutiny of the developed solution by peers within the problem domain and through the specification of the developed solution prototype, ensuring that the artifact was rigorously defined, coherent, and internally consistent with industry requirements. Established development and testing techniques were used throughout.
Guideline 6: Design as a search process Design must utilize available means to reach desired ends while satisfying laws in the problem domain.	The method of artifact was closely aligned to industry inputs and resources in use, enabling the solution to be constructed according to the problem definitions and within the constraints (economic, biological and other concerns) of the industry under consideration.
Guideline 7: Communication of Research Solution must be presented effectively to both technology-oriented and management-oriented professionals.	This was achieved through system demonstrations and evaluations by target users and stakeholders within the target industry. Both technical and business-relevant evaluation criteria were provided in documents for practitioners and industry experts. The solution artifact was also presented at an academic conference.

Appendix B

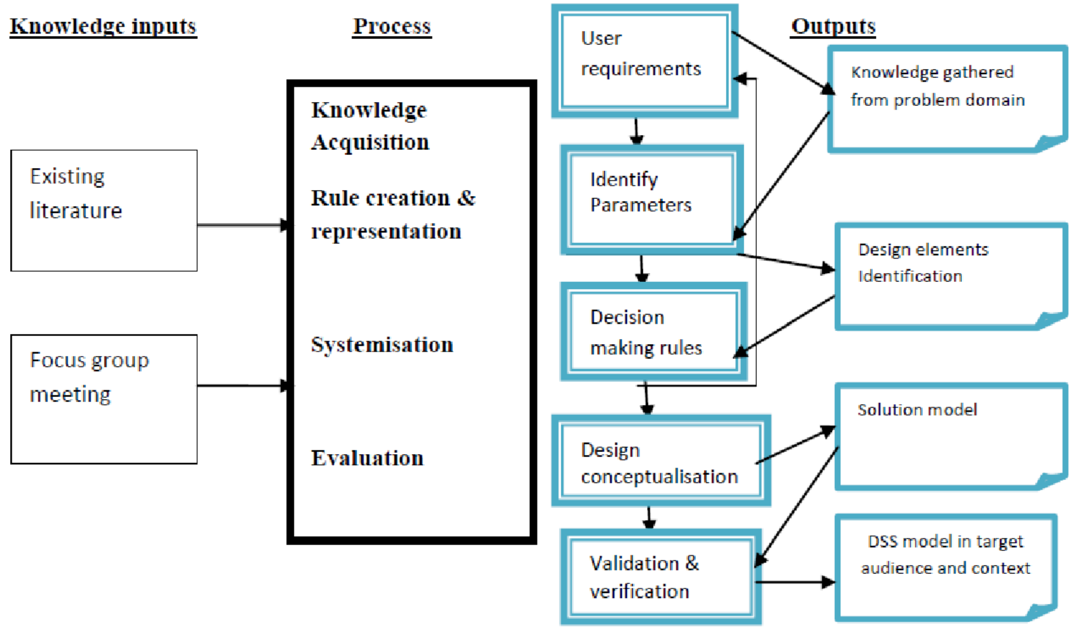


Figure B1: Methodology for DSS Solution Development in Forestry Pest Management (Partly Adapted from Purao and Storey, 2008)

About the Authors

Shah Jahan Miah is an associate professor of information systems at the College of Business, Victoria University, Melbourne. He has also held academic positions at the University of the Sunshine Coast, Griffith University, and James Cook University, Australia. Since receiving his PhD from Griffith University in the area of business decision support systems, his research interests have expanded to include business intelligence and big-data analytics. He now serves on the editorial boards of *International Journal of Business Intelligence Research, Journal of Education and Information Technologies,* and *Australasian Journal of Information Systems* and has led and edited special issues for each. Dr Miah has produced over 120 fully refereed publications including research books, journal articles, book chapters, and international conference articles. His work has appeared in top-tier outlets of the information systems field, such as *Information and Management, Knowledge-Based Systems, Telematics and Informatics, Information Technology & People,* and *Information Systems*.

John Gammack is a former professor of information systems in the College of Technological Innovation at Zayed University, Abu Dhabi. Dr. Gammack received his PhD on the topic of expert knowledge acquisition and modeling from Cambridge University, where he studied at the MRC Applied Psychology Unit. Dr. Gammack's primary field of research is knowledge management and decision support, particularly in the social, cultural, and human aspects of their application areas. He is also interested in other aspects of informatics and has published many books and articles in this general area. Prior to working at Zayed University, he worked as an academic in China, the UK, and Australia, where he was department head and chair at two Universities.

Judy McKay is a former professor of information systems in Swinburne Business School at Swinburne University of Technology, Melbourne. She researches sociotechnical aspects of requirements analysis, IT implementation, and design science. She has published in journals such as *Journal of Information Technology, Information Technology and People, Journal of Information Technology Teachings Cases, Communications of the Association of Information Systems*, and *Australasian Journal of Information Systems*. Since retiring from full-time employment, she works as a coach for Research Coaching Australia, and serves on the board of a private educational institution.