8-6-2011

Toward a Maturity Model for DSS Development Processes

Omar El-Gayar
Dakota State University, omar.el-gayar@dsu.edu

Amit V. Deokar
Dakota State University, Amit.Deokar@dsu.edu

Jie Tao
Dakota State University, jtao16065@pluto.dsu.edu

Toward a Maturity Model for DSS Development Processes

Omar F. El-Gayar
College of Business and Information Systems
Dakota State University
omar.el-gayar@dsu.edu

Amit V. Deokar
College of Business and Information Systems
Dakota State University
amit.deokar@dsu.edu

Jie Tao
College of Business and Information Systems
Dakota State University
jtao16065@pluto.dsu.edu

ABSTRACT

Despite recent progress with Decision support systems (DSS) development methodologies, a gap still exist in terms of the ability to assess the maturity of an organization with respect to its DSS development process. A need exist to be able to describe DSS development processes at a meta-level. Equally important is the ability to provide organization with prescriptions to increase the maturity of their DSS development processes.

In this paper, we propose a Decision Support System Maturity Model (DSS-MM). The model draws on extant literature related to DSS development methodologies, practices and processes to identify pertinent DSS development practices and define maturity models for these practices. From a theoretical perspective, this research presents the first maturity model specifically targeting DSS development. From a practical perspective, the model provides a framework for organizations to assess their DSS development maturity level and devise process improvement initiatives to address any limitations with existing practices.

Keywords

Decision support systems, development processes, development methodologies, maturity model.

INTRODUCTION

Decision support systems (DSS) (Eom, 1999; Gerrity, 1970; Keen & Morton, 1978; Power, 2002) aim to support decision makers by providing access to knowledge and data pertinent to the decision situation under consideration. Over the years, these systems were commonly referred to using a variety of names depending on the approach for decision support, e.g., model-based, knowledge-based, communication-based, and data-based DSS. These systems have also been referred collectively as Decision Making Support System (DMSS) (Forgionne, Cervantes-Perez, & Gelman, 2010).

Analogous to information systems in general, improving development processes for DSS has been a recurrent concern (Gachet & Haettenschwiler, 2006). Saxena (1991) argues that the development of DMSS has often turned out to be innovative but ad hoc, which is in the low level of maturity. Inspired by the advances of information technologies, the development methodologies have been improved in the same time. However, due to the nature of DMSS (high complexity and uncertainty), the development as well as the implementation of the DMSS, regardless of the type and purpose, has been proved to be very time and resources consuming. Accordingly, a large number of DSS development processes have been proposed in the literature (Mora T., et al., 2010; Mora T. et al., 2006). These processes varied significantly in their approach. For example, while some processes focused on the decision support function, others have approached DSS development from a strictly software/systems engineering perspective. Recently, (Gachet & Haettenschwiler, 2006) proposes a 'tripatriate' approach attempts to integrate among these perspectives. In conjunction to research focusing on the system development, a stream of implementation research attempts to identify critical success factors for DSS (Alavi & Joachimsthaler, 1992; Clark, Jones, & Armstrong, 2007). Regardless, of the emphasis of the development methodology or the findings of the implementation research, a gap still exist in terms of the ability to assess the maturity of an organization with respect to its DSS development process. A need exist to be able to describe DSS development processes at a meta-level. Equally important is the ability to provide organization with prescriptions to increase the maturity of their DSS development processes.

In that regard, a maturity model (MM) is a set of structured levels for describing the extent of the process that an organization can used for creating an outcome (Paulk, Curtis, Chrissis, & Weber, 2002). MM could be used as the benchmark for the usability and effectiveness of a development process in a software engineering project (Gupta, 2009; Parthasarathy & Ramachandran, 2008; Wangenheim et al., 2010). Analogous to software engineering and other maturity models, a maturity model can potentially provide a framework for describing the maturity of DSS development process at a meta-level as well as provide guidance to organization for improving their DSS development maturity.

Accordingly, in this paper we propose a maturity model, which is appropriate to support the overall DMSS development process, regardless of the specific development methodology. The model is analogous to the Software Engineering Institute's (SEI) Capability Maturity Model's (CMM) (Paulk, 1995) five maturity levels. However, we draw on extant literature related to DSS development methodologies, practices and processes, as well as existing maturity models to identify pertinent DSS development practices and define maturity models for these practices. From a theoretical perspective, this research presents the first maturity model specifically targeting DSS development practices. From a practical perspective, the proposed model provides a framework for organizations to assess their DSS development maturity level and devise process improvement initiatives to address any limitations with existing practices. Along these lines, it can also provide advantages that were demonstrated with SEI's CMM in terms of providing a project manager of a development team with the ability to manage and allocate resources and budget, while also ensuring quality check on the product being developed. Also, accounting for organizational and social issues related to the development processes as well as pre and post development processes such as needs assessment, cost/benefit analysis, user involvement, and training can benefit the management of the overall DMSS development project.

The article is structured as following: the next section is the literature review, in which we briefly discuss major DMSS development methodologies. We also provide a brief discussion of maturity model, their purpose, and a synopsis of such models in other domain. In Section 3, we present the Decision Support System Maturity Model (DSS-MM) representing the main contribution of this paper. Next we present a case study to demonstrate the applicability of the proposed model. Within the case study we map one of the existing methodologies to the proposed model to test its usability and adaptability as the guidelines for supporting DMSS development. The paper ends with the conclusion and discussion for future research.

LITERATURE REVIEW

DMSS Development Methodologies

Many scholars have discussed the term "methodology" in the literature from a system development standpoint. Maddison et al. (1984) define a methodology as "a recommended collection of philosophies, phases, procedures, rules, techniques, tools, documentation, management and training for developers of information systems." In a similar vein, Avison and Fitzgerald (2002) describe a methodology as "a collection of procedures, techniques, tools and documentation aids which will help the systems developers in their efforts to implement a new information system. A methodology will consist of phases, themselves consisting of sub-phases, which will guide the system developers in their choice of the techniques that might be appropriate at each stage of the project and also help them plan, manage, control and evaluate information systems projects". We use the term "methodology" in a broad sense, as described above.

Over the past four decades, DMSS research has garnered significant interest. During that time, a great amount of development methodologies have been proposed in the literature. Most of the early development methodologies were based on the Software Development Life Cycle (SDLC). These methodologies could be grouped into "traditional" category according to Mora T. et al. (2010). Within this category, Arinze (1991) has analyzed major DMSS development methodologies in the earliest twenty years of DMSS history from 1970 to 1990 using three criteria: paradigm, structure and orientation. Within this representative survey, ten major DMSS development methodologies are examined. The analysis led to a contingency model for DSS methodology selection, which is a remarkable progress in the DMSS history and still seems to be illustrative recently. Similarly, Arnott (1998) provided a comparative analysis on 12 DMSS development methodologies. He suggests a comprehensive framework for DMSS evolution, which is useful for the developers of DMSS in predicting the upcoming activities in the developing process and determining which tools and methodologies to be selected. Another important approach within this category is the DSS Development Phases proposed by Turban and Aronson (1997). Some other methodologies have been provided using alternative perspectives, e.g., the nature of the Decision Problems (Meador & Mezger, 1984; Mistree, Hughes, & Bras, 1993).

A number of DSS researchers have claimed that the traditional SDLC based methodologies are not adequate for DMSS development. Specifically, DSS development is heavily intertwined with the underlying decision making processes and is significantly plagued with uncertainty and the lack of well-defined requirements. Accordingly, iterative/prototyping methods are recommended in DMSS development (Gachet & Haettenschwiler, 2006). The iterative development process implies the rapid functional releases that are frequently revised utilizing user feedback. The process continues until the final version of the system is acceptable to the users. The approach is remarkably suitable for DMSS development and obtained significant applications (Baldwin, Allen, & Ridgway, 2010; Kastner et al., 2010). Similarly, Zuubier et al.(1994) propose making use of the process models for DMSS design and development that originated from prototyping. As the technology advanced, other software engineering (SE) based development methodology have been employed into the DMSS field, such as the RUP (Rational Unified Process). Brandas (2007) has provided a conceptual framework of DMSS development approach based applying RUP, and describes the development process within the framework.

But the iterative methodologies have their own disadvantages: the challenging management requirements and risky development environment. The first important methodology is the DSS Design Cycle, proposed by Keen and Morton (1978), and emphasized by Gachet (2006). This methodology focuses on the relationships between the DSS functionalities and the practical implementations. Another significant integrated method is the Decision Support Engineering suggested by Saxena (1991), which underlines the negotiation between the developers and the end-users. The third important approach is the IDSS-M methodology by Mora T. et al.(2010). The IDSS-M method is based on Saxena's (1991) method and the DSS Development Phases method, it resembles the previous two methods and some weaknesses within the two prior methods have been improved in the IDSS-M method.

From a different perspective, implementation research seeks to identify critical success factors (CSFs) or the regrettable avoidances in the development process. Hung et al. (2007) have examined dozens of DMSS relevant articles and have summarized the crucial variables for developing a successful DMSS. According to their findings, those factors are categorized into two classes, the dependent ones versus the independent ones; the former corresponding to the user satisfactions and the latter related to the decision supporting performance. Other contributions along these lines include (Arnott & Pervan, 2008; Elbeltagi, McBride, & Hardaker, 2005; Webb & Yadav, 2003).

Prior studies listed above have depicted a comprehensive view of the evolutionary paths of DMSS development methodologies (shown in Table 1), those studies still have some limitations. Firstly, none of the methodologies seems comprehensive enough to capture all DSS development practices and process deemed critical to the successful development and deployment of such systems. Secondly, the studies do not seem to provide organizations with the ability to assess their level of maturity with respect to the adoption of any of the proposed best practices for DSS development. Thirdly, coupled with the lack of assessment of current maturity is the lack of ability to provide prescriptions for organizations to improve their DSS development methodology. The next sub-section presents a brief review of maturity model and their applications in software engineering and other areas.

Table 1. DMSS development methodologies in different categories

Category	Development Methodology	Source
SDLC (System Develop Life Cycle)	Arinze reported 10 methodologies Arnott examined 12 methodologies Sage's SDLC-based DMSS development method	(Arinze, 1991) (Arnott, 1998) (Sage, 1991)
Iterative Development Methodologies	Turban & Aronson's method Zuubier's method	(Turban & Aronson, 1997) (Zuurbier, et al., 1994)
Integrated Methodologies	Keen & Morton's method Decision Supporting Engineering Mora's I-DMSS methodologies	(Keen & Morton, 1978) (Saxena, 1991) (Mora T., et al., 2010)
End-user oriented methodologies	End-user programming language Mistree et al.'s methodology Igbaria et al.'s methodology	(Meador & Mezger, 1984) (Mistree, et al., 1993) (Igbaria & Guimaraes, 1994)
CSF-based methodologies	Arnott's Eight key issues DSS usage factors Webb's qualify factors	(Arnott & Pervan, 2008) (Elbeltagi, et al., 2005) (Webb & Yadav, 2003)

Maturity Models

The concept of maturity was originally used in Quality Management field (Crosby, 1979). The SEI first introduced maturity model (MM) in the software engineering domain. The SEI uses the CMM to measure the achievement of certain capability. According to Kaner's definition of CMM (Kaner & Karni, 2004), CMM could be used as guidelines for selecting process improving strategies, thus, it is applicable for supporting the develop process of DMSS.

Paulk et al. (1995) suggests using CMM as guidelines for improving software processes. The capabilities of an organization are evaluated then categorized at different levels. The traditional levels within CMM include: initial, repeatable, defined, manageable and optimizing. The lower levels mean weak capabilities while the higher levels are corresponded to the capabilities with higher maturity(M.C. Paulk, et al., 2002). The CMM has been applied in many fields within the Software Engineering (Dounos & Bohoris, 2010; Gupta, 2009; Sivakumar, Abrahams, Hogg, & Hartley, 2010; Wangenheim, et al., 2010). Meantime, CMM is also applied in other fields that are relevant to the organizational capabilities.

As mentioned in the earlier section, the early development methods are ad hoc. As such, they are located in the lower level within the CMM. In his article, Kaner and Karni (2004) uses an expanded CMM to support DMSS develop process and then achieves remarkable results. But their research is primarily focused on the DMSS related to the knowledge management, as we mentioned above, the knowledge-based DMSS is just one type of DMSS, so that it makes their study less universal.

A MATURITY MODEL FOR DSS DEVELOPMENT PROCESSES

In building a maturity model for development processes involved in building DSS, relevant process categories and associated levels need to be articulated. Toward this end, we draw on extant literature discussed in previous sections related to DSS development methodologies, practices and processes, as well as existing maturity models.

DSS-MM is organized around four major process categories or sections as shown in Figure 1. These categories represent technical as well as social processes that DSS development typically involves. For organization purposes, the first three process categories have been presented in a manner similar to the components described in Gachet and Haettenschwiler's (2006) tripartite DSS development approach, the fourth category captures the social and organizational processes related to DSS development.

<u>Section I. Integrated Decision-Making and Software Engineering Focused Processes</u>
- I.A Decision task analysis practices
- I.B Functional requirements analysis practices

<u>Section II. Software Engineering Focused Processes</u>
- II.A Requirements gathering practices
- II.B System design practices
- II.C Prototyping practices
- II.D Evaluation practices
- II.E Reusability practices

<u>Section III. Decision-making Focused Processes</u>
- III.A Paradigm-specific (model-driven DSS illustrated here) knowledge base development practices
- III.B Extensibility practices
- III.C Reusability practices

<u>Section IV. Organizational processes</u>
- IV.A DSS cost and benefit analysis
- IV.B Organizational readiness and user/executive commitment to the DSS
- IV.C User involvement in development
- IV.D DSS training

Figure 1. Architecture of Decision Support Systems Maturity Model (DSS-MM)

Level	**Generic Definition**
Level 1	Some awareness, with some activities underway related to the practice, not necessarily at the right time.
Level 2	General awareness, activities related to the practice conducted in a relatively timely manner but following an informal or ad hoc approach.
Level 3	Systematic approach/methodology adopted in all activities relevant to the practice, with timely execution, but low on details, and effectiveness.
Level 4	Rigorous and timely employment of systematic approach/methodology in all activities relevant to the practice, ensuring good effectiveness.
Level 5	Well-defined, and rigorous approach followed in a timely manner in all activities relevant to the practice, along with ongoing assessment and feedback to ensure sustainability and continuous improvement.

Figure 2. Generic maturity level definitions

<table>
<tr><td colspan="7">Section, Group # and Name – Brief description of the group of practices within the Section. For example, I.A pertains to decision task analysis practices.</td></tr>
<tr><td colspan="2">Diagnostic Questions</td><td colspan="5">1.0 General questions relative to this group of practices to help assess current capability
2.0</td></tr>
<tr><td rowspan="2">DP#</td><td rowspan="2">Development Practices</td><td colspan="5">Capability Levels</td></tr>
<tr><td>Level 1</td><td>Level 2</td><td>Level 3</td><td>Level 4</td><td>Level 5</td></tr>
<tr><td rowspan="5"></td><td>Define the specific practice</td><td colspan="5">Statements describing each of the levels relative to the practice under consideration
(Indicate the current level by circling a “C” and the desired level by circling a “D”)</td></tr>
<tr><td></td><td>C D</td><td>C D</td><td>C D</td><td>C D</td><td>C D</td></tr>
<tr><td>Evidence</td><td colspan="5">Supportive data or observations utilized in assessing the current capability level (indicated by C)</td></tr>
<tr><td>Desired Maturity indicators</td><td colspan="5">Outcomes and behaviors that will be demonstrated as the desired level of maturity (indicated by D) is attained</td></tr>
<tr><td>Action Plan</td><td colspan="5">Indicates the action plan in moving toward the desired capability level</td></tr>
</table>

Figure 3. DSS-MM maturity matrix template

Within each of these categories, a number of practices have been identified that are likely to be of concern to the DSS development team or organization. While these practices are not meant to be all-inclusive, the focus is on highlighting significant behaviors and concerns related to DSS development processes. A development team’s progress toward building a DSS may be assessed using this subclass of key practices that can provide a preview of how well the overall DSS development process is advancing. The project manager responsible for overseeing the DSS development project may perform such an assessment.

At the core of the DSS-MM are different maturity matrices. Five maturity statements are developed for each practice within the four process categories, ranging from least capable (Level 1) to superior (Level 5). Generic definitions or characterizations of each capability level, as shown in Figure 2 have been developed which serve as a reference in developing maturity statements for each of the practices included in DSS-MM. A template capturing the standard format for organizing the information in DSS-MM has been designed, as shown in Figure 3. The segments shown in italics are completed during the assessment exercise. Also, two levels, “C” and “D” indicate the choices to represent the “current” or “desired” capability levels. In the following subsections, we describe each of the process categories within DSS-MM.

DSS-MM-Section I. Integrated Decision-Making and Software Engineering Focused Processes

Processes within this category are at the higher level of abstraction as compared to those focused either on software engineering focused issues or decision-making focused issues, both of which are discussed in subsequent DSS-MM sections. Their significance is that they are related to generic tasks such as decision task analysis, problem space definition/complexity, and functional requirements analysis, and have practical implications in terms of how software engineering and decision-making processes are oriented to meet the overall goals of the DSS (P. G. W. Keen & Scott Morton, 1978; Saxena, 1991; Sprague & Carlson, 1982) as well as DSS quality (Clark, et al., 2007). From a technical standpoint, these processes may be encapsulated within a controller-like module, such as the DSS kernel conceived by Gachet and Haettenschwiler (2006), that acts as an intermediary and interfaces with the other categories of processes. Some questions that relate to these integrated processes include the following:

- Decision task analysis practices
  - Have artifacts, such as decision scenarios, been developed to elicit decision problem requirements?
  - Have decision problem characteristics such as problem space complexity been inferred through decision problem requirements gathering process?
  - Has task structure been formally elicited through user interactions and documented using techniques such as cognitive maps, influence diagrams or decision situation diagrams?
- Functional requirements analysis practices
  - Has support analysis or functionality analysis been conducted (such as through walkthrough of the decision scenarios) to gain knowledge about users expectations of decision “support” functionalities/operations?
  - Has technology assessment of the DSS objectives been conducted from feasibility standpoint and priorities?

An example of integrated processes as discussed above is practice I.A.1. *Elicit and identify decision problem requirements*. The five maturity levels for this practice are:

Level 1 – Decision problem requirements have been identified late in the DSS development process
Level 2 – Decision problem requirements have been identified earlier in the system development process, however requirements are gathered in an ad hoc manner without rigorous user input, and documented in a non-systematic manner
Level 3 – Decision problem requirements have been identified early on in the development process with rigorous user input and documentation (such as through decision scenarios), but without much detail.
Level 4 – Decision problem requirements have been identified early on in the development process with rigorous user input and systematic and detailed documentation (such as through decision scenarios).
Level 5 – In addition to meeting level 4 standards, decision problem requirements are continually assessed during the development process with user and executive involvement.

DSS-MM-Section II. Software Engineering Focused Processes

Processes that fall under this category are grounded in software and systems engineering fields. These processes are primarily concerned with the system aspect of the DSS and have little, if any, focus on decision-making aspects of the DSS. SDLC (Sage, 1991), prototyping (Alavi, 1984), and end-user development (Alavi & Weiss, 1985; Kreie, Cronan, Pendley, & Renwick, 2000) are examples of approaches that have much relevance to these processes. Given the inherent differences in various software engineering approaches, a "one size fits all" model to describe key practices applicable to all approaches is infeasible. Each of these approaches has slightly different concerns to be addressed and best practices to be followed. Despite these differences, some representative diagnostic questions that relate to the software engineering focused processes have been identified from extant literature which include:

- Requirements gathering practices
  - Has user analysis been conducted to identify the users' prior experience with such systems as well as their expectations (e.g., learning time, user-driven interaction)?
  - Have technology (hardware/software) specific requirements identified?
  - Have requirements been elicited and/or identified with respect to the user interface (e.g., interface style, core features, adaptability features)?
  - Have requirements and goals been identified from a usability standpoint (e.g., training time, performance effectiveness, flexibility support)?
- System design practices
  - Have system design specifications been developed (e.g., system architecture, design artifacts such as use cases, class diagrams, sequence diagrams)?
  - Have user interfaces been modeled?
- Prototyping practices
  - Does the prototyping follow a pre-specified strategy (e.g., one shot, evolutionary)?
- Evaluation practices
  - Are user evaluations planned/conducted to incorporate user feedback? If so, how frequently (what stages)?
  - Are system validation and/or testing strategies in place for evaluation?
- Reusability practices
  - Is the system design amenable to incorporate different decision-oriented processes?
  - Other reusability related practices pertain to security, data/model persistence, system maintenance, user management.

DSS-MM-Section III. Decision-making Focused Processes

Processes focused on decision-making are vital part of DSS, and as such have been the focus of study particularly in the early periods of DSS research (Blanning, 1979; Martin, 1982; Stabell, 1983). These processes are centered on the design and development of "knowledge bases", consisting of essential data and information (e.g., schemas and/or instances of data, models, inference rules, cases, ontologies) as well as operations for manipulating this data and information, to ultimately provide decision support to the end-user (Gachet & Haettenschwiler, 2006). Reflecting on prior DSS research, these processes were also central to the concepts of DSS generators (Sprague & Carlson, 1982) and general problem processing system (GPPS) (Bonczek, Holsapple, & Whinston, 1982). Given that that there different types of DSS (data-driven, model-driven, document-driven, etc.), knowledge base development practices will be somewhat different for each DSS type. Also, typically, decision-making processes have to take into account extensibility and reusability requirements as well. Some representative diagnostic questions that relate to decision-making focused processes for model-driven DSS (Krishnan & Chari, 2000; Muhanna & Pick, 1994; D. J. Power & Sharda, 2007) include:

- Paradigm-specific (model-driven DSS illustrated here) knowledge base development practices
    - Has a systematic decision model analysis been conducted to identify a suitable modeling technique (e.g., decision analytic techniques such as AHP, math programming, simulation, statistical, economic)
    - In case of model-driven DSS, have model management requirements such as model-data independence, and model-solver independence been identified?
- Extensibility practices
    - Are there mechanisms in place to allow extending the knowledge base to incorporate variations and extensions documented via decision task analysis discussed in DSS-MM-Section-I?
- Reusability practices
    - Are the access and manipulation operations for information (data, models, rules, ontologies, etc.) in the knowledge base structured to allow different clients or interfaces to be linked to the core knowledge base?

DSS-MM-Section IV. Organizational processes

Along with processes pertaining to DSS system development, related organizational or social processes also play a key role in successful development and deployment of DSS. Clark et al. (2007) and Alavi and Joachimsthaler (1992) have reviewed the literature and synthesized a number of such factors relevant to DSS implementation and adoption that form the basis of section of DSS-MM. While some practices precede the actual DSS building efforts (e.g., cost/benefit analysis), others are concurrent to the DSS development, and yet others follow the development process (e.g., training). Some example questions to be assessed within category of processes include:

- DSS cost and benefit analysis
    - Has a systematic cost/benefit analysis been conducted prior to engaging the DSS building?
- Organizational readiness and user/executive commitment to the DSS
    - Does the DSS solve a business need as perceived by the user(s) and executive(s)?
    - Has decision support business strategy been aligned with technology strategy?
    - Has the organizational readiness (including factors such as the culture surrounding the decision-making process, the use of DSS applications, organizational attitude toward decision support and process improvement) been assessed?
    - Have the developers worked closely with executives in the development of DSS to ensure a broader support?
- User involvement in development
    - Have the developers worked closely with the users in the development of DSS? At what phases?
    - How effective is the communication link between the users and developers/analysts?
- DSS training
    - To what extent is training in the DSS technology base (hardware and software) provided to the users?
    - To what extent is training the decision structure for a given problem space provided to the users?

APPLICATION OF DSS-MM TO DSS DEVELOPMENT METHODOLOGIES

Table 2 shows a mapping between DSS-MM and a commonly referenced Decision Support Engineering (DSE) methodology, proposed by Saxena (1991). While there is a significantly close mapping between the Sections I, II and III of DSS-MM, the DSE methodology cannot be mapped to the organizational processes mentioned in Section IV of DSS-MM because of the lack of this consideration. We have conducted similar mapping to other methodologies (not presented here due to lack of space) such as IDSSE-M (Mora T., et al., 2010), tripartite approach (Gachet & Haettenschwiler, 2006), and the design cycle (P. Keen & Morton, 1978) with analogous observations.

Table 2: Mapping between DSS-MM and Decision Support Engineering Methodology (Saxena, 1991)

DSS-MM Practices	Saxena's (1991) DSE Methodology
Section I: Integrated Decision-Making and Software Engineering Focused Processes	
Decision task analysis practices	Decision task analysis
Functional requirements analysis practices	Requirements engineering: (b) support analysis

Section II: Software Engineering Focused Processes	
Requirements gathering practices	Requirements engineering: (a) user analysis; (f) user interface analysis; (g) hardware/software environment; (h) usability analysis
System design practices	DSS design: (b) user interface modeling; (d) designing DSS architecture
Prototyping practices	Prototyping
Evaluation practices	User Evaluation
Reusability practices	-
Section III: Decision-making Focused Processes	
Paradigm-specific (model-driven DSS illustrated here) knowledge base development practices	Requirements engineering: (c) decision model analysis; (d) knowledge base analysis; (e) database analysis;
Extensibility practices	Requirements engineering: (g) hardware/software environment
Reusability practices	DSS design: (a) decision and knowledge base modeling; (c) database modeling

CONCLUSION

In this paper we present DSS-MM, a maturity model for decision support systems development process. The model is based on extant literature pertaining to DSS development methodologies and practices, DSS implementation research, as well existing maturity models. From a theoretical perspective, this research presents the first maturity model specifically targeting DSS development processes and practices. From a practical perspective, the proposed model provides a framework for organizations to assess their DSS development maturity level and guidance for further improving upon these practices.

The development of the maturity model was guided by the key DSS development methodologies from the literature. This approach has the limitation of possibly excluding certain key elements that may not be represented formally in the existing methodologies, but may play an important role in the overall DSS development process. We propose to address this issue as explained below.

The research can be extended along a few dimensions. We propose to refine the maturity model further in an iterative manner till a saturation of key aspects of the development process has been attained. An expert analysis study based on the Delphi technique is currently underway, which will lead to a refined model whose content will be validated by experts in the field. As part of future work, validation of the proposed model can be further enhanced through a case study project aimed at studying the applicability of the model in a ‘real-life’ DSS development environment.

REFERENCES

1. Alavi, M. (1984). An assessment of the prototyping approach to information systems development. *Communications of the ACM, 27*(6), 556-563.
2. Alavi, M., & Joachimsthaler, E. A. (1992). Revisiting DSS implementation research - A meta-analysis of the literature and suggestions for researchers. *MIS Quarterly, 16*(1), 95-116.
3. Alavi, M., & Weiss, I. R. (1985). Managing the risks associated with end-user computing. *Journal of Management Information Systems, 2*(3), 5-20.
4. Arinze, B. (1991). A contingency model of DSS development methodology. *Journal of Management Information Systems, 8*(1), 149-166.
5. Arnott, D. (1998). *A framework for understanding decision support systems evolution.* Paper presented at the Proceedings of the 9th Australasian Conference on Information Systems.
6. Arnott, D., & Pervan, G. (2008). Eight key issues for the decision support systems discipline. *Decision Support Systems, 44*(3), 657-672.
7. Avison, D. & Fitzgerald, G. (2002). *Information Systems Development: Methodologies, Techniques, and Tools* (3rd ed.). The McGraw-Hill Companies.
8. Baldwin, J. S., Allen, P. M., & Ridgway, K. (2010). An evolutionary complex systems decision-support tool for the management of operations. *International Journal of Operations & Production Management, 30*(7-8), 700-720. doi: 10.1108/01443571011057308

9. Blanning, R. W. (1979). The functions of a decision support system. *Information & Management, 2*(September), 71-96.
10. Bonczek, R. H., Holsapple, C. W., & Whinston, A. B. (1982). The evolution from MIS to DSS: Extension of data management to model management. In M. J. Ginzberg, W. R. Reitman & E. A. Stohr (Eds.), *Decision Support Systems* (pp. 61-78). Amsterdam, North-Holland.
11. Brandas, C. (2007). Unified approach in the DSS development process. *Informatica, 41*(1), 98-102.
12. Clark, T. D., Jones, M. C., & Armstrong, C. P. (2007). The dynamic structure of management support systems: Theory development, research focus, and direction. *MIS Quarterly, 31*(3), 579-615.
13. Crosby, P. B. (1979). *Quality is free*: McGraw-Hill New York.
14. Dounos, P., & Bohoris, G. (2010). *Factors for the Design of CMMI-Based Software Process Improvement Initiatives*.
15. Elbeltagi, I., McBride, N., & Hardaker, G. (2005). Evaluating the factors affecting DSS usage by senior managers in local authorities in Egypt. *Journal of Global Information Management, 13*(2), 42-65.
16. Eom, S. (1999). Decision support systems research: current state and trends. *Industrial Management & Data Systems, 99*(5), 213-221.
17. Gachet, A., & Haettenschwiler, P. (2006). Development processes of intelligent decision-making support systems: Review and perspective. In J. N. D. Gupta, G. A. Forgionne & M. Mora T. (Eds.), *Intelligent Decision-making Support Systems: Foundations, Applications, and Challenges* (pp. 97-121). London: Springer-Verlag.
18. Gerrity, T. (1970). *The design of man-machine decision systems*: Massachusetts Institute of Technology.
19. Gupta, R. (2009). *A maturity model for CBSE.* Paper presented at the Proceedings of the 2nd India software engineering conference, Pune, India.
20. Hung, S., Ku, Y., Liang, T., & Lee, C. (2007). Regret avoidance as a measure of DSS success: An exploratory study. *Decision Support Systems, 42*(4), 2093-2106.
21. Igbaria, M., & Guimaraes, T. (1994). Empirically testing the outcomes of user involvement in DSS development. *Omega-International Journal of Management Science, 22*(2), 157-172.
22. Kaner, M., & Karni, R. (2004). A capability maturity model for knowledge-based decisionmaking. *Information, Knowledge, Systems Management, 4*(4), 225-252.
23. Kastner, M., Li, J., Lottridge, D., Marquez, C., Newton, D., & Straus, S. E. (2010). Development of a prototype clinical decision support tool for osteoporosis disease management: a qualitative study of focus groups. *BMC Medical Informatics and Decision Making, 10*. doi: 4010.1186/1472-6947-10-40
24. Keen, P., & Morton, M. (1978). *Decision support systems: an organizational perspective* (Vol. 35): Addison-Wesley Reading, MA.
25. Keen, P. G. W., & Scott Morton, M. S. (1978). *Decision Support Systems: An Organizational Perspective*. Reading, MA: Addison-Wesley Publishing Company.
26. Kreie, J., Cronan, T. P., Pendley, J., & Renwick, J. S. (2000). Applications development by end-users: Can quality be improved? *Decision Support Systems, 29*(2), 143-152.
27. Krishnan, R., & Chari, K. (2000). Model management: Survey, future research directions and a bibliography. *Interactive Transactions of OR/MS, 3*(1).
28. Maddison, R., Baker, G., Bhabuta, L., Fitzgerald, G., Hindle, K., Song, J., Stokes, N. & Wood, J. (1984) Feature Analysis of Five Information Systems Methodologies. In T. Bemelmans (Ed.), *Beyond Productivity: Information Systems Development For Organizational Effectiveness*, North Holland: Elsevier Science Publishers.
29. Martin, M. P. (1982). Determining information requirements for DSS. *Journal of Systems Management, December*, 14-21.
30. Meador, C., & Mezger, R. (1984). Selecting an end user programming language for DSS development. *MIS Quarterly, 8*(4), 267-281.
31. Mistree, F., Hughes, O., & Bras, B. (1993). The compromise decision support problem and the adaptive linear programming algorithm. *Structural Optimization: Status and Promise, 50*, 251-290.
32. Mora T., M., Forgionne, G., Cervantes-Perez, F., & Gelman, O. (2010). IDSSE-M: Intelligent decision support systems engineering methodology. In L. C. Jain & C. P. Lim (Eds.), *Handbook on Decision Making: Techniques and Applications* (Vol. 1, pp. 29-53). Berlin: Springer-Verlag.
33. Mora T., M., Forgionne, G., Gupta, J. N. D., Garrido, L., Cervantes, F., & Gelman, O. (2006). A strategic descriptive review of the intelligent decision-making support systems research: the 1980-2004 period. *Intelligent Decision-making Support Systems*, 441-462.
34. Muhanna, W. A., & Pick, R. A. (1994). Meta-modeling concepts and tools for model management: A systems approach. *Management Science, 40*(9), 1093-1123.
35. Parthasarathy, S., & Ramachandran, M. (2008). Requirements engineering method and maturity model for ERP projects. *International Journal of Enterprise Information Systems, 4*(4), 1.

36. Paulk, M. C. (1995). *The Capability Maturity Model: Guidelines For Improving The Software Process* (Vol. 66). Reading, MA: Addison-Wesley.
37. Paulk, M. C., Curtis, B., Chrissis, M. B., & Weber, C. V. (2002). Capability maturity model, version 1.1. *Software, IEEE, 10*(4), 18-27.
38. Power, D. (2002). *Decision support systems: concepts and resources for managers*: Praeger Publishers.
39. Power, D. J., & Sharda, R. (2007). Model-driven decision support systems: Concepts and research directions. *Decision Support Systems, 43*(3), 1044-1061.
40. Sage, A. P. (1991). *Decision Support Systems Engineering*. New York, NY: Wiley.
41. Saxena, K. B. C. (1991, Jan 8-11). *Decision support engineering: A DSS development methodology.* Paper presented at the Proceedings of the Twenty-Fourth Annual Hawaii International Conference on System Sciences (HICSS-24 '91).
42. Sivakumar, G., Abrahams, F., Hogg, K., & Hartley, J. (2010). *SOI (Service Oriented Integration) and SIMM (Service Integration Maturity Model): An Analysis.* Paper presented at the Proceedings of the 2010 6th World Congress on Services.
43. Sprague, R. H., & Carlson, E. D. (1982). *Building Effective Decision Support Systems*. Englewood Cliff, NJ: Prentice Hall.
44. Stabell, C. B. (1983). A decision-oriented approach to building DSS. In J. L. Bennett (Ed.), *Building Decision Support Systems* (pp. 221-260). Reading, MA: Addison-Wesley Publishing Company.
45. Turban, E., & Aronson, J. (1997). *Decision Support Systems and Intelligent Systems*: Prentice Hall PTR Upper Saddle River, NJ, USA.
46. Wangenheim, C. G. v., Hauck, J. C. R., Zoucas, A., Salviano, C. F., McCaffery, F., & Shull, F. (2010). Creating software process capability maturity models. *IEEE Software, 27*(4), 92-94. doi: 10.1109/ms.2010.96
47. Webb, H., & Yadav, S. (2003). Quality factors for DMSS assessment: an application of research frameworks *Decision making support systems: achievements, trends, and challenges for the new decade* (pp. 272).
48. Zuurbier, J., Brinkkemper, J., Offereins, M., & Odding, N. (1994, Jan 4-7). *Towards a design methodology for decision support systems.* Paper presented at the Proceedings of the Twenty-Seventh Hawaii International Conference on System Sciences (HICSS-27 '94).