

## Chapter 7.4

# Context in Decision Support Systems Development

**Alexandre Gachet**

*University of Hawaii at Manoa, USA*

**Ralph Sprague**

*University of Hawaii at Manoa, USA*

### INTRODUCTION

Finding appropriate decision support systems (DSS) development processes and methodologies is a topic that has kept researchers in the decision support community busy for the past three decades at least. Inspired by Gibson and Nolan's curve (Gibson & Nolan 1974; Nolan, 1979), it is fair to contend that the field of DSS development is reaching the end of its expansion (or contagion) stage, which is characterized by the proliferation of processes and methodologies in all areas of decision support. Studies on DSS development conducted during the last 15 years (e.g., Arinze, 1991; Saxena, 1992) have identified more than 30 different approaches to the design and construction of decision support

methods and systems (Marakas, 2003). Interestingly enough, none of these approaches predominate and the various DSS development processes usually remain very distinct and project-specific. This situation can be interpreted as a sign that the field of DSS development should soon enter in its formalization (or control) stage. Therefore, we propose a unifying perspective of DSS development based on the notion of context.

In this article, we argue that the context of the target DSS (whether organizational, technological, or developmental) is not properly considered in the literature on DSS development. Researchers propose processes (e.g., Courbon, Drageof, & Tomasi, 1979; Stabell 1983), methodologies (e.g., Blanning, 1979; Martin, 1982; Saxena, 1991; Sprague & Carlson, 1982), cycles (e.g., Keen & Scott Morton, 1978; Sage, 1991), guidelines (e.g.,

DOI: 10.4018/978-1-59904-843-7.ch011

for end-user computer), and frameworks, but often fail to explicitly describe the context in which the solution can be applied.

## BACKGROUND

ADSS is broadly considered as “a computer-based system that aids the process of decision making” (Finlay, 1994). Sprague uses a definition that indicates key components of the DSS architecture. A DSS is a “computer-based system which helps decision makers confront ill-structured problems through direct interaction with data and analysis models” (Sprague, 1980). In a more detailed way, Turban (1995) defines it as “an interactive, flexible, and adaptable computer-based information system, especially developed for supporting the solution of a non-structured management problem for improved decision making. It utilizes data, provides an easy-to-use interface, and allows for the decision maker’s own insights.” This second definition gives a better idea of the underlying architecture of a DSS. Even though different authors identify different components in a DSS, academics and practitioners have come up with a generalized architecture made of six distinct parts: (a) the data management system, (b) the model management system, (c) the knowledge engine, (d) the user interface, (e) the DSS architecture and network, and (f) the user(s) (Marakas, 2003; Power, 2002).

One section this article, Key Terms, briefly defines nine DSS development methodologies popular in the DSS literature. A typical methodology is represented by the steps in Table 1.

The exact number of steps can vary depending on the aggregation level of each phase. Moreover, steps are usually sequenced in an iterative manner, which means the process can iterate to an earlier phase if the results of the current phase are not satisfactory. Even though these processes are useful from a high-level perspective, we argue

that they poorly support the DSS designers and builders to cope with contextual issues. The next paragraphs provide a couple of examples to illustrate this argument. The first example is related to the user interface. The DSS community widely recognizes that the user interface is a critical component of a DSS and that it should be designed and implemented with particular care. But how critical is this component? On the one hand, if we consider a DSS that is intended to be used by a wide range of nontechnical users (for example, a medical DSS for the triage of incoming patients in an emergency room that will be used by nurses and MDs working under pressure), then the user interface is indeed the single most critical component of the DSS, at least from a usability/acceptability point of view. In this context, the human-computer interaction (HCI) literature tells us that usability must definitely be considered before prototyping takes place, because the earlier critical design flaws are detected, the more likely they can be corrected (Holzinger, 2005). There are techniques (such as usability context analysis) intended to facilitate such early focus and commitment (Thomas & Bevan, 1996). On the other hand, if we consider a highly specific DSS that will be handled by a few power-users with a high level of computer literacy (sometimes the DSS builders themselves), then the user interface is less critical and usability considerations can be postponed until a later stage of the development process without threatening the acceptability of the system. This kind of deci-

*Table 1. Phases of the DSS design and development life cycle (Sage, 1991)*

- |   |
|---|
| <ol style="list-style-type: none"> <li>1. Identify requirements specifications</li> <li>2. Preliminary conceptual design</li> <li>3. Logical design and architectural specifications</li> <li>4. Detailed design and testing</li> <li>5. Operational implementation</li> <li>6. Evaluation and modification</li> <li>7. Operational deployment</li> </ol> |
|---|

9 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

[www.igi-global.com/chapter/context-decision-support-systems-development/44171](http://www.igi-global.com/chapter/context-decision-support-systems-development/44171)

## Related Content

---

### Research Review: Globally Distributed Requirements Engineering and Agility

Vanita Yadav (2013). *Business Innovation, Development, and Advancement in the Digital Economy* (pp. 1-11).

[www.irma-international.org/chapter/research-review-globally-distributed-requirements/74133](http://www.irma-international.org/chapter/research-review-globally-distributed-requirements/74133)

### Adoption of a New Online Travel Management System for FED-AK

Aundrea Kell, Shari Pierreand Bogdan Hoanca (2013). *Cases on Performance Measurement and Productivity Improvement: Technology Integration and Maturity* (pp. 59-79).

[www.irma-international.org/chapter/adoption-new-online-travel-management/69107](http://www.irma-international.org/chapter/adoption-new-online-travel-management/69107)

### Mining Association Rules from XML Documents

Laura Irina Rusu, Wenny Rahayuand David Taniar (2009). *Services and Business Computing Solutions with XML: Applications for Quality Management and Best Processes* (pp. 176-196).

[www.irma-international.org/chapter/mining-association-rules-xml-documents/28975](http://www.irma-international.org/chapter/mining-association-rules-xml-documents/28975)

### Conflicts, Compromises, and Political Decisions: Methodological Challenges of Enterprise- Wide E-Business Architecture Creation

Kari Smolanderand Matti Rossi (2009). *Selected Readings on Information Technology and Business Systems Management* (pp. 71-93).

[www.irma-international.org/chapter/conflicts-compromises-political-decisions/28634](http://www.irma-international.org/chapter/conflicts-compromises-political-decisions/28634)

### Introduction

Len Aspreyand Michael Middleton (2003). *Integrative Document and Content Management: Strategies for Exploiting Enterprise Knowledge* (pp. 1-27).

[www.irma-international.org/chapter/introduction/24070](http://www.irma-international.org/chapter/introduction/24070)