Survey of DSS Development Methodologies

Natheer K. Gharaibeh
College of the Computer Science and Engineering, Taibah University,Yanbu Branch, Saudi Arabia

Abdulaziz Al-Raddadi
College of the Computer Science and Engineering, Taibah University,Yanbu Branch, Saudi Arabia

INTRODUCTION

Finding appropriate DSS (Decision Support System) development processes and methodologies (Gachet, 2005) is a topic that has kept researchers in the decision support community busy for the past three decades at least. The DSS development methodologies changed after 1995 because the DSS community has always shown great interest in the underlying technology and rapidly emerging Information Technology underpins DSS (O'Leary, 2008).

Inspired by Hayen (Hayen, 2004) it is fair to contend that the field of DSS development is reaching the end of its matured stage, which is characterized by the multiplicity of processes and methodologies in all areas of decision support. Despite that, none of these approaches predominate and the various DSS development processes usually remain very distinct. This situation can be interpreted as a sign that the field of DSS development should soon enter in its formalization (or control) stage. Therefore, the objective of this chapter is to focus on the controlled integration of the existing solutions in a unified body of knowledge and to come up with advantages and disadvantages of the previous methodologies

One of the latest studies (Gachet, 2006) showed that DSS development methodologies are characterized by different underlying philosophies, historically, a large amount of research on the development of DSS focuses on organizational issues, technical issues, or both kinds of issues at the same time. Whereas it is widely recognized that these two categories represent the dominant sources of issues that DSS builders had to overcome in the past, a third category, KM issues, gradually surfaces. Bolstered by advances in information technology in general, and artificial intelligence in particular, the field of knowledge management increases the number of development issues previously dealt with partly from an organizational perspective and partly from a technical perspective, but rarely as a perspective of its own.

In the DSS literature, experts prescribe a variety of approaches or methodologies for designing and developing DSS. Everyone does not however agree on what methodology works best for building different types of DSS. For example Gachet (Gachet, 2006) who proposed a bipartite approach in which the software engineering part is separated from the knowledge engineering part. Another example is Turban (Turban, 2005) who described a development process consisting of 6 phases for DSS constructed by end users, also there are many researchers studied DSS Development methodologies from many perspective for example (Marakas, 2003; Zarate, 1998; Elgarah, 2002) .Many researchers preview and compares the Development methodologies of DSS which will be expressed on the next section.

The purpose of this chapter is double. In the second section, it reviews the comparative studies of DSS development methodologies, the goal of this review is to give the reader a thorough understanding of the past and on-going research in DSS development, Section 3 reviews the DSS development methodologies by dividing these perspectives into: three main categories (according to the Organizational, technical and

DOI: 10.4018/978-1-4666-5202-6.ch216

people factors) Section 4 analyze and compare the Development methodologies of DSS. Section 5 shows the required Characteristics of SDM for building DSS, and a summary of the chapter is provided in section 6.

COMPARATIVE STUDIES

DSS design and development processes have been the subject of several comparative studies in the past. Each of these studies categorizes the processes according to varying criteria. For example, Arinze (Arinze, 1991) surveys ten DSS methodologies by paradigm, structure, and orientation. The paradigm refers to the models underlying the methodologies (decision-driven, process-driven, data- driven, or systemic). The structure indicates the approach used for guiding the development process (stage or contingency). The orientation involves the developmental guidelines adopted by DSS researchers (normative or descriptive). Arinze's survey leads to a research model developed and used relate the functions of DSS methodology to decision-making environments and the relevant processes within them.

Ariav and Ginzberg (1985) contend that a large number of DSS design studies have emphasized a single set of related issues such as the nature of decision situations, components, tools and technologies of DSS, the processes of DSS design and use, etc. They strongly believe that a systemic view of DSS can provide a unified approach to effective design of DSS and can serve as a basis for accumulating DSS research results. The fundamental system properties outlined by Churchman are as follows:

1. Objectives of the total system: the problem is defined and the objective of the system must be viewed in relation to the other components and to larger systems/the whole system.
2. The systems environment;
3. The resources of the system;
4. the components of the system: a system is composed of interrelated elements and the design of a system is the design of subsystems and their relationships and
5. The management of the system.

Blair, Debenham and Edwards in 1997 (1997) described a comprehensive study conducted to understand the methodologies which are being used to design DSSs and to identify the key methodological problems and benefits with using these methodologies to assist IDSS developers in understanding what support can be gained from using existing design methodologies and hence choose the correct one for their project.

Power (2000) mentioned three approaches for building DSS: systems development life cycle (SDLC) which is the most commonly encountered term used to describe the steps in a traditional systems development methodology, prototyping approach and end-user development of DSS. In both of the later two approaches a portion of the DSS is quickly constructed, then tested, improved, and expanded. Prototyping is similar to a related approach called rapid application development (RAD).

Arnott (2004) analyzes twelve DSS methodologies and DSS development use cases with a focus on DSS evolution. He proposes a framework based on the aetiology, lineage, and tempo of evolution, an etiology refers to the causes of evolution (exogenous or endogenous triggers), "lineage refers to whether evolution occurs within an application or between applications, and tempo relates to the pattern of evolution over time" (continuous evolution, punctuated equilibrium, or quantum evolution). Arnott claims that his framework clarifies the nature of DSS evolution and "may help systems analysts predict what may happen next in the development processes and help them in deciding which techniques and tools are likely to succeed with each class of evolution.," furthermore the framework and the case study findings are used to define a research agenda for evolutionary DSS development.