Decision Support Systems 68 (2014) 49–63

Contents lists available at ScienceDirect

Decision Support Systems

journal homepage: www.elsevier.com/locate/dss

Decision support in hierarchical planning systems: The case of procurement planning in oil refining industries

Kasper Bislev Kallestrup [a], Lasse Hadberg Lynge [a], Renzo Akkerman [b,*], Thordis Anna Oddsdottir [a]

[a] Department of Management Engineering, Technical University of Denmark, Produktionstorvet 424, 2800 Kgs. Lyngby, Copenhagen, Denmark
[b] TUM School of Management, Technische Universität München, Arcisstr. 21, 80333 Munich, Germany

ARTICLE INFO

Article history:
Received 3 October 2013
Received in revised form 11 August 2014
Accepted 13 September 2014
Available online 20 September 2014

Keywords:
Advanced planning systems
Hierarchical planning
Crude oil operations
Procurement planning

ABSTRACT

In this paper, we discuss the development of decision support systems for hierarchically structured planning approaches, such as commercially available advanced planning systems. We develop a framework to show how such a decision support system can be designed with the existing organization in mind, and how a decision process and corresponding software can be developed from this basis. Building on well-known hierarchical planning concepts, we include the typical anticipation mechanisms used in such systems to be able to decompose planning problems, both from the perspective of the planning problem and from the perspective of the organizational aspects involved. To exemplify and develop our framework, we use a case study of crude oil procurement planning in the refining industry. The results of the case study indicate an improved organizational embedding of the DSS, leading to significant savings in terms of planning efforts and procurement costs. In general, our framework aims to support the continuous improvement of advanced planning systems, increasing planning quality in complex supply chain settings.

1. Introduction

Managing supply chains is a complex task and is often done with hierarchically structured advanced planning systems (APSs) [47]. Due to the complexity and uncertainty in supply chains, it is normally not possible or desirable to create fully automated decision systems that rapidly identify and execute optimal decisions. Even if it were, it is likely that managers would not trust such a system blindly, due to the very high potential costs of erroneous decisions. Instead, managers turn to decision support systems (DSSs), which support planning processes as good as possible.

A DSS involves a human–computer interaction and the software part usually provides a range of information that managers use to decide on an action. Literature on DSSs in relation to APSs focuses primarily on modeling or data structure aspects. As documented by Zoryk-Schalla et al. [51], focusing on modeling aspects early in the development of DSSs for APSs, without properly defining the planning process and its characteristics, can lead to significant implementation problems. This calls for research on the development process of APSs, especially related to the planning process complexities, information requirements, as well as the organizational embedding [20].

The planning processes covered in APSs normally span different functional domains (e.g. production, distribution, sales) and include decision problems with different time horizons and granularity (from strategic to operational). This range of planning processes is often also reflected in organizational structures, which increases the need for coordination between the different processes. It is exactly this coordination in the hierarchical structure and the related information flows that are key factors in the development and implementation of APSs [48], and often the reason to implement an APS in the first place [20]. Even though there is an increasing body of work on APSs and other hierarchical planning systems, how to develop additional decision support in a structured way remains a challenge, especially when taking into account the existing planning infrastructure and its organizational embedding.

In this paper, we address this problem by developing a framework to support DSS design, considering organizational aspects and process design early in the development process. More specifically, we contribute to the decision support literature by answering (i) how a DSS for an APS can be designed with the existing organization in mind?, (ii) how a decision process can be developed from this basis? and (iii) how the information obtained from the first two steps eases and directs the development of the enabling software? As we base our work on well-known hierarchical planning concepts, we also contribute by providing insights on the anticipation mechanisms used in such systems to be able to decompose planning problems. Since we propose methods for creating DSSs in commonly used APS settings, we also ensure professional relevance [4]. Throughout the paper, we will use a case study of procurement planning in the refining industry, an industry where advanced planning systems have traditionally seen extensive use [14,40]. However, the procurement planning problem has been an underdeveloped aspect within APSs [33], and previous research has often not managed to capture

* Corresponding author.
E-mail address: renzo.akkerman@tum.de (R. Akkerman).

http://dx.doi.org/10.1016/j.dss.2014.09.003
0167-9236/

the decision problem properly [35], and it is therefore a good environment for the discussion of the DSS development process.

In the following section, we first discuss the main literature streams related to our study. Based on this, Section 3 develops an initial development framework for model-based DSSs for APSs. In Section 4, we then outline the general structure of procurement planning in the refinery industry, followed by the application of our development framework to create a DSS in Section 5. In Section 6, we further refine our framework based on the case results, followed by conclusions and further research directions in Section 7.

2. Related literature

2.1. Decision support systems

Creating DSSs is a way for organizations to improve the efficiency and effectiveness of their decision processes. Power [38] argues that how to improve a decision process depends on how ill-structured the decision problem is. Ill-structured problems are defined by what they are not, namely well-structured problems. A well-structured problem is a problem that is routinely carried through, where the solution can be checked and where the end goal can be met in a reasonable amount of time [46]. According to Power [38], problems that are well-structured can normally be automated in a decision system, while ill-structured problems need special decision studies. Problems that lie in-between these extremes (semi-ill-structured problems) can normally benefit from a DSS.

Conceptually, a DSS is a formalization of the knowledge and experience held by employees involved in the process, structured in a way that enhances the decision maker's ability to choose the best solution to a complex problem [25]. According to Holsapple and Whinston [16], a DSS is a human–computer interaction, where the computer consists of four elements, illustrated in Fig. 1.

A DSS may use a database analysis system to create metadata that can give new insights for managers or provide a mathematical model of the problem domain that greatly reduces a problem's solution domain. Early descriptions of management information systems used for decision support focused on the analysis of the decision system and condensation of data [1] as a means to ensure that software systems would provide only information that was relevant to the decision maker. Courbon et al. [8] described the design of such systems as evolutionary, a concept that was expanded and popularized by Keen [23]. Keen argued that a DSS can only exist if it is a product of an adaptive process between the system and the user, where the system encourages the user to take new approaches, which in turn lead the user to request more features from the system. Both Keen and Courbon et al. focused on the software engineering aspects of DSSs, assuming that a good decision process would evolve naturally. Other authors, including Blanning [6] and Linger and Burstein [31] used the analysis of the decision making process to identify what functions a DSS should have. Blanning also mapped modules of the DSS to the different departments in an organization.

Gachet and Haettenschwiler [12] reviewed nine different DSS development methodologies and advocated for those that combine system engineering aspects and decision making process aspects. These integrated approaches include Keen and Scott-Morton's [22] widely

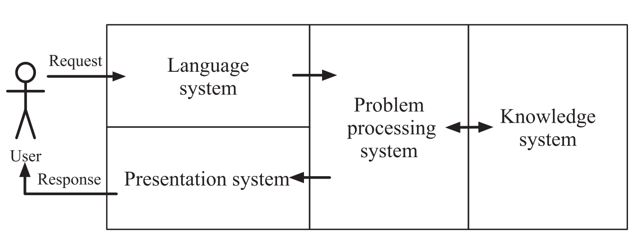

Fig. 1. Structure of DSSs according to Holsapple and Whinston [16].

accepted "Design Cycle" and Saxena's [41] "Decision Support Engineering", both of which place the decision analysis task or process development task as an antecedent of the software development task. Saxena's approach also includes a range of interactions that challenge the sequential structure, in order to allow for the evolutionary characteristics of DSSs.

2.2. Hierarchical planning structures

Hierarchical planning is important if one seeks to optimize systems where scheduling is critical and non-trivial tactical decisions, such as procurement and network planning, have a larger horizon than it is possible to optimize scheduling problems for. Planners usually solve this problem by splitting the system into at least two planning levels, an aggregate level and a detailed level. One of the earliest approaches to structure a planning system hierarchically was presented by Hax and Meal [15]. They proposed a hierarchical system that can "make decisions in sequence, with each set of decisions at an aggregate level providing constraints within which more detailed decisions must be made". Bitran et al. [5] also argued that hierarchical planning models can replace difficult-to-solve stochastic planning models when demand for individual products is stochastic but deterministic for aggregated product families. A robust aggregate plan is defined by Lasserre and Mercé [29] as a plan for which a feasible disaggregation policy can be formulated. Gfrerer and Zäpfel [13] subsequently discussed how robust aggregate plans can be enforced through top-down coordination.

A conceptual framework for describing hierarchical structures, like those in hierarchical planning systems, was proposed by Schneeweiss [42]. In a hierarchical structure, there will be a top level (aggregate level) and a base level (detailed level). The top level can instruct the base level and the base level can react to these instructions. In top-down hierarchical structures like APSs, the base level cannot present a reaction before the decision is implemented in the object system (production system), but instead, the top level anticipates how the base level will react. The anticipation in the top level can then be updated/refined based on ex-post feedback from the object system. In subsequent work, Schneeweiss [43] distinguishes three categories of anticipation: (i) considering characteristics of the base level directly (perfect anticipation), (ii) considering approximations of base-level characteristics (approximate anticipation), or (iii) considering part of the base-level characteristics (implicit anticipations). For non-perfect anticipation, which is normally the case in practice, the reaction from the object system can be quite different from the anticipations. Decision-makers can reduce the problem by increasing the quality of the anticipation, which is often a key aspect in the design and operation of hierarchical planning systems.

Hierarchical models can in some simple cases be autonomous, but in practice, a hierarchical planning system takes the form of an APS that combines autonomous decision modules with DSS modules [47]. The use of DSS modules is necessitated by the complexity of supply chains and, to some extent, the capacity of modern algorithms and computers. APS modules are either technically integrated or integrated through organizational processes. Most of the DSSs in APSs are model-based optimization tools (using Alter's [2] taxonomy), simply because planning and scheduling have traditionally been areas that heavily utilize operations research methodology.

Researchers report that APSs are widely used in the industry, but it has been argued that this is not reflected in DSS research [32]. As a consequence, DSS literature does not always describe the task of positioning a DSS, which determines the users, their objective and their organization. In a single-level planning process, this may be straightforward but for APSs, where multiple planning levels exist and many organizational units are involved, positioning a DSS is not straightforward.