

The evolution of a production planning system: A 10-year case study

Kenneth N. McKay^a, Gary W. Black^{b,*}

^aDepartment of Management Sciences, University of Waterloo, Waterloo, Ont. N2L 3G1, Canada

^bCollege of Business, University of Southern Indiana, Evansville, IN 47712, USA

Received 13 April 2006; received in revised form 17 January 2007; accepted 5 February 2007

Available online 23 March 2007

Abstract

This paper describes the evolution of a production planning system (PPS) from a simple work sequence generation tool to a useful, sustained scheduling system. Three stages of evolution are described. In the first stage, a Gantt chart sequencing tool was converted to a scheduler's information system. This change was driven by the need to support the scheduler's daily task. The second stage of evolution was caused by an MRP–ERP conversion. The integration and conversion increased overhead and complexity in the job task and hence the tool, including the transformation of the previously integrated dispatching/scheduling task into separate dispatching and scheduling activities. The third stage of evolution has been small continuous improvements driven by management reporting requirements. PPS was developed in 1996 and has been fully operational since January 1997. Two major insights are discussed in this paper: the implications of supporting the scheduling task versus work sequence generation, and the software design requirements for evolutionary change as the software is used in an ever-changing situation.

© 2007 Elsevier B.V. All rights reserved.

Keywords: Production planning system; Decision support system; Enterprise resource planning; Materials requirements planning; System evolution

1. Introduction

Production planning systems (PPS) are a specialized form of decision support system (DSS). There are over 100 commercial systems available [1]. The purpose of a PPS is to take manufacturing requirements, match them with a model of the factory or the supply chain and, using various algorithms and technology, and craft a work sequence either automatically or with manual intervention. In this paper, the term scheduling will be reserved for what the human scheduler does as their daily job task—part of which is the creation of detailed work sequences. This differentiation is important since one of the major drivers of the software evolution described in this paper was the explicit recognition and support of the scheduling task, above and beyond the creation of detailed sequences. McKay and Wiers [2,7] describe the production control tasks of planning, scheduling, and dispatching. In this case study the

individuals with a title of *scheduler* performed the tasks of scheduling and dispatching.

If detailed work sequencing is focused upon, the problem appears relatively simple and many sequencing tools have been quickly built with a Gantt chart interface and some form of sequencing engine. The concentration has been on the data representation, database, Gantt chart visualization and manipulation, and on the sophisticated mathematics used to create a recommended sequence. Unfortunately, there are few success stories and there are still many challenges that remain when creating planning and logistics systems that have to work in a real factory [3–6]. While there is not one sole issue or reason for PPS success or failure [7], one potential issue relating to the failure of planning systems is the distinction between *generating a sequence* and *scheduling* and the lack of information system support for the latter. This was the first major insight obtained from this longitudinal case study. Another possible reason for system failure is the rigid structure or inability to easily adapt to changing requirements in the factory environment—both information system and organization changes. Learning what aspects of the software should be easily changed was the second major insight.

* Corresponding author. Tel.: +1 812 461 5385.

E-mail addresses: kmckay@uwaterloo.ca (K.N. McKay), GBlack@usi.edu (G.W. Black).

This case study will describe three stages of evolution are described:

- moving from a basic work sequence generation tool to a scheduler's information system—driven by the need to support the scheduler's task;
- moving from a legacy MRP system to ERP (SAP)—driven by changes in the information system infrastructure;
- continuous evolution—driven by management and organizational requirements.

The first evolution occurred during a 4-month period following the initial live test of the software in the Fall of 1996. This evolution saw the code triple in size and the functionality more than double. This basic system was then used for approximately 1.5 years before the ERP migration. When the plant converted to the ERP, the scheduler's tool doubled in complexity to deal with the added functionality required by the ERP interface and associated task changes. The decision support system continues to be used and has been fully operational and in daily use for over 10 years. The factory using the software has been participating in ongoing research on production control throughout this time period. This relationship has presented a relatively unique research opportunity to research, develop, deploy, and evolve a live decision support system over an extended time horizon.

The first phase of evolution is perhaps of the most interest to researchers who are contemplating the creation of decision support tools for a scheduler or dispatcher task. The lessons and insights gained relate to the understanding of the task and software architecture. The second phase of evolution is more interesting to individuals integrating existing production control tools with legacy or ERP systems. The insights in this context relate more to the changes and impositions created by the larger and more rigid systems. The third phase of evolution provides lessons to both audiences—the need for change and adaptation to the organizational demands as a plant itself evolves.

This paper is organized as follows. Section 2 will present a review of DSS evolutionary literature followed by specific discussion of case studies addressing evolution in the production planning and scheduling field. Section 3 will provide background information for the case study. Section 4 will discuss the system development processes and underlying architecture. Sections 5–7 will discuss in-depth each of the three evolutionary stages. Section 8 will discuss and summarize the results. Section 9 will present some concluding thoughts.

2. Evolution of decision support systems

Research reporting case studies on the evolution of decision support systems is relatively sparse. Evolutionary development in decision support was first hinted at by Meador and Ness [8] and Ness [9] as part of their 'middle-out' design. This design was in response to the 'top-down' versus 'bottom-up' methodology debate at that time. Courbon et al. [10] provided the first general statement of DSS evolutionary development.

He argued that development processes are not implemented in a linear or parallel fashion, but rather in continuous action cycles involving significant user interaction. As each cycle is completed, the system gets closer to its final state.

Keen [11] advanced Courbon's work to develop a model for understanding the dynamics of DSS evolution. The approach was termed 'adaptive design' and was based upon cyclic interactions between each pair of three basic elements: the builder/systems analyst, the user and the system. Courbon [12] later described these cycles as sequences of 'action' (i.e., when the designer releases a new version and the user works with it) and 'reflection' (i.e., feedback to the designer). He concluded that DSS evolution is best conceptualized as a learning process.

In an analysis of system adaptation and evolution, Sprague and Carlson [13] identified four levels of DSS flexibility: the flexibility to solve a problem, the flexibility to modify the DSS to handle different problems, the flexibility to adapt to major changes and the flexibility to evolve with changes in technology. They believed these levels exist in a hierarchy with technology-based evolution at the top. They argued that "DSS must evolve or grow to reach a 'final' design because no one can predict or anticipate in advance what is required. The system can never be final; it must change frequently to track changes in the problem, user and environment because these factors are inherently volatile".

Arnott [14] discussed the nature of the DSS evolutionary process and presented a framework for defining it based on tempo, lineage and etiology. He contended the dominant tempo of DSS evolution was punctuated equilibrium in which logic and operation alternated between periods of being relatively static and periods of rapid change. This finding was in contrast to previous theories citing continuous evolution as the dominant tempo [10–12]. Lineage was discussed in terms of within-application and between-application evolution. Etiology was discussed in terms of cognitive causal factors (e.g., system use, training, interaction with analysts/peers/consultant) versus environmental causal factors (e.g., technology change, personnel change, industry change). By combining the various lineages and etiology, a framework was presented with four major classes of evolution along with the most likely tempo(s) for each class. A large DSS case study pertaining to a large semi-government building project was used to validate and expand the framework. Finally, the framework and case study findings were used to define a research agenda for evolutionary DSS development. This agenda was comprised of a set of 11 research questions within four different categories: evolutionary tempo, etiology, methodology and technology, and the nature and role of people involved.

The above approaches and theories illustrate the importance of the concept of evolutionary development to DSS theory and practice. The notion that DSS's evolve through an iterative process of system design and usage has been central to the theory since the inception of the field [14]. While existing work on the DSS development describes the process in terms of a final system resulting from an adaptive process of user/analyst learning and system change, the real nature of DSS evolution may be even more complex. To this end, case study papers

متن کامل مقاله

دریافت فوری ←

ISIArticles

مرجع مقالات تخصصی ایران

- ✓ امکان دانلود نسخه تمام متن مقالات انگلیسی
- ✓ امکان دانلود نسخه ترجمه شده مقالات
- ✓ پذیرش سفارش ترجمه تخصصی
- ✓ امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
- ✓ امکان دانلود رایگان ۲ صفحه اول هر مقاله
- ✓ امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
- ✓ دانلود فوری مقاله پس از پرداخت آنلاین
- ✓ پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات